Mobility in Wireless Sensor Networks: Advantages, Limitations and Effects

A thesis submitted for the Degree of
Doctor of Philosophy of
The Australian National University

Oday Jerew
July 2011

This document was produced using TEX , LATEX and BIBTEX

Declaration

The contents of this thesis are the results of original research and have not been submitted for a high degree to any other university or institution.

Much of the work in this thesis has been published or has been submitted for publication.

Conference Papers

- Oday D. Jerew, Haley M. Jones, and Kim L. Blackmore, On the Minimum Number of Neighbours for Good Routing Performance in MANETs, In *Proc. of International Conference on Mobile Ad Hoc and Sensor Systems (MASS09), IEEE*, pages 12–15, October, 2009.

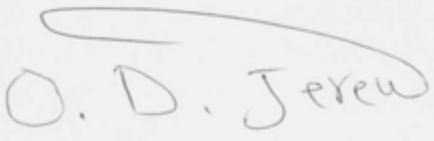
- Oday Jerew and Weifa Liang, Prolonging Network Lifetime Through the Use of Mobile Base Station in Wireless Sensor Networks. In *Proc. of International Conference on Advances of Mobile Computing and Multimedia (MoMM), ACM*, pages 170–178, December, 2009.

Journal papers

- Oday Jerew, Kim Blackmore and Weifa Liang, Mobile Base Station and Clustering to Maximize Network Lifetime in Wireless Sensor Networks. IEEE Transaction on Wireless Communications (to be submitted).

- Oday Jerew and Kim Blackmore, Multipath Routing Scheme For Mobile Relays in Wireless Sensor Networks. IEEE Transaction on Wireless Communications (to be submitted).

- Oday Jerew and Kim Blackmore, Estimation of Hop Count in Multihop Paths in Sparse and Dense Ad Hoc and Sensor Networks. IEEE Transaction on Vehicular Technology (to be submitted).



Oday Jerew

Acknowledgements

Without the support of the many faces in my life, this work would not have been possible. I would like to acknowledge and thank each of the followings:

- First of all, I would like to show my deep appreciation for my advisor, Dr. Kim Blackmore, for her guidance, support, encouragements, and patience. Her deep insight brought me a lot of ideas that were invaluable for all of my work.
- I would like to thank the amazing Dr.Tony Flynn, for his helpfulness, support and patience, not to mention his sense of humour and optimism. This thesis would certainly not been possible without his support.
- I would like to thanks Dr. Haley Jones and Dr. Weifa Liang for their helpful ideas, insight, feedback and general rock-solid reliability as a supervisor.
- I would like to thank the Iraqi Ministry of Higher Education and Scientific Research to provide me with the opportunity and financial to come to ANU and conduct research work.
- My thanks to the Department of Engineering and Information Technology in ANU for their supportive and the use of their facilities in the production of this thesis.
- Last, but definitely not least, I would like to thank my parents: mum and dad and my wife for their moral support, love and for always being there when I need them.

Abstract

The primary aim of this thesis is to study the benefits and limitations of using a mobile base station for data gathering in wireless sensor networks. The case of a single mobile base station and mobile relays are considered.

A cluster-based algorithm to determine the trajectory of a mobile base station for data gathering within a specified delay time is presented. The proposed algorithm aims for an equal number of sensors in each cluster in order to achieve load balance among the cluster heads. It is shown that there is a tradeoff between data-gathering delay and balancing energy consumption among sensor nodes. An analytical solution to the problem is provided in terms of the speed of the mobile base station. Simulation is performed to evaluate the performance of the proposed algorithm against the static case and to evaluate the distribution of energy consumption among the cluster heads. It is demonstrated that the use of clustering with a mobile base station can improve the network lifetime and that the proposed algorithm balances energy consumption among cluster heads. The effect of the base station velocity on the number of packet losses is studied and highlights the limitation of using a mobile base station for a large-scale network.

We consider a scenario where a number of mobile relays roam through the sensing field and have limited energy resources that cannot reach each other directly. A routing scheme based on the multipath protocol is proposed, and explores how the number of paths and spread of neighbour nodes used by the mobile relays to communicate affects the network overhead. We introduce the idea of allowing the source mobile relay to cache multiple routes to the destination through its neighbour nodes in order to provide redundant paths to destination. An analytical model of network overhead is developed and verified by simulation. It is shown that the desirable number of routes is dependent on the velocity of the mobile relays. In most cases the network overhead is minimized when the source mobile relay caches six paths via appropriately distributed neighbours at the destination.

A new technique for estimating routing-path hop count is also proposed. An analytical model is provided to estimate the hop count between source-destination pairs in a wireless network with an arbitrary node degree when the network nodes are uniformly distributed in the sensing field. The proposed model is a significant improvement over existing models, which do not correctly address the low-node density situation.

Abbreviations

AODV	Ad Hoc On-Demand Distance Vector
BFS	Breadth First Search
BS	Base Station
CDF	Cumulative Distribution Function
CEDAR	Core Extraction Distributed Ad Hoc Routing
CGSR	Cluster-head Gateway Switch Routing
DSDV	Destination-Sequenced Distance-Vector
DSR	Dynamic Source Routing
GPS	Global Positioning System
LCA	Linked Cluster Algorithm
LEACH	Low-Energy Adaptive Clustering Hierarchy
MAHSN	Mobile Ad Hoc Sensor Network
MR	Mobile Relay
PC	Personal Computer
PDA	Personal Digital Assistants
PDF	Probability Density Function
RREP	Route Reply
RREQ	Route Request
RCH	Real Cluster Head
RS	Real Segment
TORA	Temporally Ordered Routing Algorithm
UAV	Unmanned Aerial Vehicles
VCH	Virtual Cluster Head
VS	Virtual Segment
WLAN	Wireless Local Area Network
WRP	Wireless Routing Protocol
WSN	Wireless Sensor Network
ZRP	Zone Routing Protocol
$Cov(p, r)$	A circular region centred at position p with radius r
Area(.)	Area
$[.]$	Average Operator
$\lceil . \rceil$	Ceiling Operator
$E\{.\}$	Expectation Operator
$\lfloor . \rfloor$	Floor Operator
$\max\{.\}$	Maximum Operator
$\min\{.\}$	Minimum Operator
Pr	Probability

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Thesis Motivation

The research described in this thesis has been largely motivated by the perceived need for understanding of the effect of the use of mobility in Wireless Sensor Networks (WSNs) on network performance. The potential benefits and limitations of using single and multiple mobile entities for data gathering has only recently been fully recognised [12][13].

Data gathering in WSNs is one of the most frequent and fundamental operations, and requires the sensor nodes to monitor the sensing field for as long as possible. As sensor nodes have limited energy resources and are powered by small batteries, energy consumption is a critical issue in the design of WSNs that effect network lifetime.

Some WSNs use mobility to prolong network lifetime by allowing a mobile base station (BS) to roam a sensing field and gather data from sensor nodes through a short transmission range. The energy consumption of each sensor node is then reduced, since fewer relays are needed for the sensor node to relay its data packet to the BS. As the speed of mobile BS is very slow compared with the speed of data packets which travel in multi-hop forwarding, the increased latency of data gathering when employing mobile BS presents a major performance bottleneck. Thus, the time a mobile BS takes to tour a large sensing field may not meet the stringent delay requirements inherent in some mission-critical, real-time applications. Therefore, planing the trajectory and determining the speed of mobile BS need to be considered in order to achieve the delay requirements.

Planning the moving tour of a mobile BS is a critical issue in the maximization of network lifetime and meeting data-gathering delay requirements. In order to maximize network lifetime, a mobile BS collects data using single-hop commu-

nication, which requires a long time since it has to visit each sensor node. On the other hand, using multi-hop communication to reduce data-gathering time increases sensor-node power consumption and thus shortens network lifetime.

Data-gathering time could also be decreased by increasing the speed of the mobile BS. However, the contact time between the mobile BS and sensor nodes is decreased and thus increases the probability of packet losses. In addition, the network lifetime and number of packet losses increase when the network scale is expanded. Thus the use of a mobile BS is not sufficient to achieve the required network performance of a large scale network.

The moving of sensor nodes has been introduced in the literature in order to achieve requirements such as improve converge and connectivity or improve network lifetime by moving to a new location to help bottleneck sensors by inheriting sensing, transmission and receiving responsibilities. Several algorithms for planning the trajectory of the mobile BS have been proposed to prolong network lifetime, which do not analyse the limitations when the network scale is expanded and investigate the effect of BS velocity on packet losses.

Multiple mobile entities, which we refer to as mobile relays (MRs) are proposed to roam the sensing field and buffer sensing data to be forwarded to the BS. MRs are used to improve network lifetime as the network scale is expanded. However, little attention is paid to the routing between MRs. Most of the literature assumes that MRs can interact with each other directly in order to send sensing data to the BS, where this can be achieved either by restricting the mobility area of MRs which adds more constraint to trajectory planning or by assuming the MRs have unlimited energy resources to reach each other and the BS. Thus, there is a need for further investigation into the effect of the mobility of MRs on routing performance as the mobility inevitably incurs additional overhead in data communication protocols, whose overhead can potentially offset the benefit brought by mobility.

The mobility of sensor nodes in the sensing field forms mobile ad hoc networks. Therefore, in the following section, we introduce the main properties of ad hoc networks, mobile ad hoc networks and then sensor networks.

1.2 Ad Hoc Networks

A wireless ad hoc network is a collection of nodes with no pre-established infrastructure. Each node has a wireless communication capability to communicate with others. Since there is no central entity in ad hoc networks the nodes must

participate in order to organise themselves into a network. Ad hoc networks show a distinct departure from traditional infrastructure wireless networks such as cellular networks and WLANs, in that there is no need for a central access point or BS. In ad hoc networks, nodes that are within each other's transmission range can communicate directly and are responsible to discover each other. These nodes are often energy constrained, that is, batteries are the main energy resource with a great diversity in their capabilities. Therefore, the transmission range of nodes is limited. Intermediate nodes work as routers in order to relay data packets between nodes that are not lie within each transmission range. That is, data packets need to be delivered over a path involving multiple nodes (multi-hops).

1.2.1 Wireless Signal Propagation Model

In wireless ad hoc networks, electromagnetic radio waves are used for communication. When radio waves travel through media which contains many objects, they experience several propagation mechanisms such as reflection, diffraction and scattering. The wireless channel (transmission medium) is susceptible to a variety of transmission impediments such as path loss, interference, and blockage. These factors restrict the range, data rate, and the reliability of the wireless transmission and places fundamental limitations on the performance of wireless communication systems [14]. Therefore the transmission range of nodes varies in time and space. This i referred to as the physical layer in the Open Systems Interconnection (OSI) reference model (a standard model used to describe computer network architecture).

The data link layer of the OSI model is responsible for ensuring reliable frame communication by detecting and possibly correcting errors that may occur in the physical layer. The network layer then uses these frames to generate packets and is concerned with routing the packets to their destination [15]. In this thesis we ignore physical layer effects, and instead deal with the network layer by considering the routing path of data packets such that the power consumption of wireless nodes is minimized. Therefore, we assume in Chapters 3, 5 and 6 signal attenuation is due only to path loss related to distance transmitted.

1.3 Mobile Ad Hoc Networks

Nodes in ad hoc network could be cellular phones, personal digital assistants (PDAs), pocket PCs and laptops. These nodes are mobile and have to join or leave the network when they move arbitrarily, this resulting in rapid and

unpredictable topology changes. In this energy-constrained, dynamic, multi-hop environment, nodes need to organise themselves dynamically in order to provide the necessary network functionality in the absence of fixed infrastructure or central administration. Thus a mechanism for path identification and maintenance is needed.

Due to node mobility and variation in transmission-range power, mobile ad hoc networks experience a high level of topology variability. All the nodes must participate to provide network functions, such as data forwarding and routing activities, in order to self-organize network topology. Node mobility strongly influences the performance of the network. Therefore, an efficient routing protocol is needed to improve network performance such as route delay, loop free routing, control overhead, scalability and power conservation.

1.3.1 Routing Protocols For Mobile Ad Hoc Networks

Ad hoc wireless network-routing protocols can be classified into three major types based on the routing information update mechanism as follows [15]:

1. Proactive (table-driven) routing protocols: Every node in this type of protocol maintains the network topology information in the form of routing tables by continuously evaluating the routes within the networks, so that when a packet needs to be forwarded, the route is already known and can be immediately used. This has the advantage that when a route is needed, the delay before packets can be sent is very small. However, it needs some time to converge to a steady state which can cause problems when the topology is changing frequently. Typical proactive routing protocols include: Destination-Sequenced Distance-Vector Routing (DSDV), Cluster-head Gateway Switch Routing (CGSR) and the Wireless Routing Protocol (WRP).

2. Reactive (On-demand) routing protocols: A node in this reactive routing protocol obtains the necessary path to destination when it is required, by using a type of global-search procedure. Thus, these protocols do not exchange routing information periodically. This form of routing may suffer a long delay since a route to destination needs to be acquired before sending a data packet. Dynamic Source Routing (DSR), Ad Hoc On-Demand Distance Vector (AODV) and Temporally Ordered Routing Algorithm (TORA) are examples of on-demand routing protocols.

   Any on-demand routing protocol must utilise some type of routing cache in order to avoid the need to rediscover each routing decision for each individual packet. One of the critical factors in an on-demand routing protocol is the

setting of the cache timeout value, since the route cache may rely on links between nodes that are no longer within wireless transmission range of each other. A large route cache timeout causes some stale information to be employed degrading network performance rather than improving it. On the other hand, small route cache timeout cause a number of valid routes to be removed before they expire and hence are an inefficient use of cache information.

3. Hybrid routing protocols: This type combines the features of proactive and reactive routing protocols. Often nodes in a network are divided into routing zones based on particular geographic regions. Routing within the same zone is implemented based on proactive routing, while reactive routing is used for routing among nodes that belong to different zones. The Linked Cluster Algorithm (LCA), Core Extraction Distributed Ad Hoc Routing (CEDAR) and Zone Routing Protocol (ZRP) are examples of this type of routing protocol.

1.3.2 Effect of Mobility on Routing Performance

Mobility in ad hoc networks is one of the most challenging in the design of routing protocols, since nodes can roam sensing fields independently of each other at varying velocities. In general, most of the literature that studies the effect of mobility in ad hoc networks shows that network performance is degraded due to link failures, that cause a significant number of routing packets to discover new paths, leading to increased network congestion and transmission latency.

Intensive research has been done on the effect of mobility on routing protocols and compares their performance using different routing-protocol metrics. Metrics used to evaluate the performance of routing protocols include [16, 17, 18, 19]:

- Throughput: Throughput measures the effectiveness of the network in delivering data packets. That is, the amount of data packets that is successfully transferred over a period of time.
- Packet delivery ratio: The ratio of the number of packets received to the number of packets sent.
- Routing overhead: The number of routing control packets requested when a data packet is successfully delivered to the destination.
- End-to-end delay: The average time difference between the time a packet is sent from the source and the time it is successfully received by the destination.

The on-demand routing protocols such as DSR and AODV perform better than the proactive, such as DSDV at high mobility rates, while DSDV perform quite well at low mobility rates [17][20], since the proactive routing protocols update the routing table whenever the network topology changes. Thus, proactive protocols are not suitable for mobile ad hoc networks in which the network topology changes frequently [21]. In addition, the performance also differs for on-demand routing protocols, for example, using the packet delivery ratio and end-to-end delay as performance metrics, DSR outperforms AODV in less demanding situations, while AODV outperforms DSR in heavy traffic load and high mobility, while, the routing overhead of DSR is lesser than that of AODV [18][20]. This is because many of their routing mechanics are different. In particular, DSR uses source routing, whereas AODV uses table driven routing framework and destination sequence numbers.

Exploring the manner in which mobility affects network communication can help the design of an efficient routing protocol. Several routing-protocol schemes have been designed that rely on identification of stable links in the networks by assuming nodes perform online measurements [22][23]. The stable links are then preferentially used for routing.

Investigating the effect of mobility using mobility metrics is necessary to measure the reliability of individual paths and discover long-lived routes. Routing protocols based on mobility metric [24, 25, 26] have shown there is an improvement in network performance such as packet delivery ratio and network overhead when mobility prediction metric is used. The prediction metric is used to predict the duration of time that two nodes remain connected.

Link residual time is a mobility metric that is used to measure the time during which two nodes are within transmission range of each other. The time until the route breaks, path residual time, can then be measured, where the reliability of a path depends on the availability of all links constituting the path. In this research, path residual time is used to decide when a path is broken and a new route request needs to be initiated.

1.3.3 Effect of Node Density

Node density, which is the number of nodes in a unit area, is another factor that effects network performance. As the network nodes are deployed randomly without any wired infrastructure and communicate via multi-hop wireless links, node density effects the connectivity of the nodes in the network. Network connectivity can be increased by increasing the number of nodes (for a fixed network area). However, increasing the number of nodes tends to reduce the effective bandwidth available for each node due to increased competition for bandwidth. In addition,

it increases the traffic load, contention and packet collision between neighbour nodes. On the other hand, when the number of nodes is small, the network may not be fully connected and therefore some nodes cannot send packets to certain destinations [27]. Increasing the transmission power of a node can achieve a higher transmission range and therefore nodes can reach more nodes via a direct links. In contrast, a node that uses a very low transmission power may become isolated without any link to other nodes. Thus, the network connectivity depends on both node density and their transmission range [28].

Connectivity is often associated with the number of neighbours (node degree). Many investigations have been conducted for the evaluation of the minimum number of neighbours needed for full connectivity in a wireless network [8, 29, 30, 31, 32] . It was first proposed by Kleinrock and Silvester in [8] that six was the 'magic number', i.e., on average every node should connect itself to its six nearest neighbours, and various papers since then have argued for magic numbers between five and eight [33, 7, 34, 35]. In this research, we use node degree as a measure of node density (rather than the number of nodes in a unit area), since it reflects the number of nodes that can be accessed using the maximum transmission range.

The node degree has a significate effect of the number of hops between the source and destination nodes. As we will demonstrate in Chapter 5, the number of hops is approximately proportional to the separation distance for very low or very high node degree, but significantly greater for node degree close to the 'magic number' values. In general, a path with a high number of hops increases the end-to-end delay and wastes the bandwidth.

1.4 Wireless Sensor Networks

Sensor networks are a special category of ad hoc wireless networks that include sensor nodes which are tiny devices that have the capability of sensing physical parameters, processing the data gathered, and communicating over the network to send data to the monitoring station (sink or BS). A sensor network is a collection of a large number of sensor nodes that are deployed in a particular region. Fig. 1.1 illustrates a traditional homogeneous wireless sensor network with flat architecture, where all nodes are equipped with identical battery capacity and hardware complexity, except for the sink node as the gateway to communicate with end users across the Internet [1].

Some of the domains of application for sensor networks are military, health care, home security, and environmental monitoring. Some of the issues that distinguish the sensor networks category of ad hoc wireless networks are as

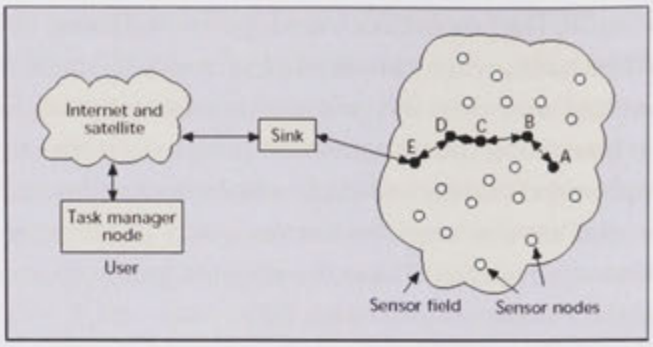


Figure 1.1: Sensor networks [1].

follows:

- Mobility of nodes: Nodes in wireless sensor networks are not assumed to be fully mobile, enabling all or a subset of nodes to provide stationary sensing abilities.
- Size of the network (scalability): The network size of sensor networks is much larger than that in ad hoc networks.
- Node density: The node density in sensor networks is larger than in ad hoc networks, which offer a small number of hops between the source and destination in sensor networks.
- Power constraints: Nodes are considered to be highly energy constrained as battery reserves are not easily replenished.
- Data sink, In general, nodes in wireless sensor networks send data packets (sensing data) to a sink, while in ad hoc networks any two nodes could be the source and destination of data packets.

The topology of WSNs is variable due to both mobility of a subset of nodes and node failures (due to energy issues). The goal of WSNs is to reduce the energy consumption of sensor node, in order to prolong its lifetime. Protocols must be designed which enable power conservation at the expense of degradation in throughput and delay characteristics.

A flat network architecture for WSNs leads to several challenges in terms of routing design, energy conservation and network management. Therefore, a hierarchical sensor network architecture is introduced, in the following section.

1.4.1 Hierarchical Sensor Network Architecture

Energy efficiency and scalability are the greatest challenges in the design of sensor networks. Therefore, hierarchical sensor network architecture is often employed in which sensors are organized into clusters with a cluster head in each cluster. Cluster heads collect sensing data and make routing and scheduling decisions.

Hierarchical sensor networks can be classified into two broad types; homogeneous and heterogeneous sensor networks [36]. In homogeneous networks all the sensor nodes are identical in terms of battery energy and hardware complexity. Some of the nodes are selected to serve as cluster heads. However, cluster heads consume more energy than other sensor nodes and as a result the cluster head fails before other nodes. Rotating the role of cluster head randomly and periodically over network nodes can help to balance energy consumption among cluster heads and hence increase the overall cost of the entire sensor network [37]. However, dynamically selected cluster heads can incur a high overhead due to frequent exchange of control packets among sensor nodes [2].

On the other hand, in a heterogeneous sensor network, one or more different types of nodes in addition to sensor nodes with different energy resources and functionality are used. The basic idea behind that is more complex hardware and extra battery energy can be embedded in cluster head nodes and can help to reduce the hardware cost of the rest of the resource limited basic sensor nodes. For instance, a two-tier hierarchical sensor network is shown in Fig. 1.2 where two types of sensor nodes are deployed in the sensing area, basic sensor nodes with limited communication capability that are mainly used for sensing the environment and sensor nodes with more powerful transceivers and batteries that act as cluster heads. The cluster head organizes basic sensor nodes into clusters, gathers sensing data and then forwards these to the BS. Clustered sensor networks can be classified as single-hop and multi-hop. A single-hop network is one in which sensor nodes use single-hop communication to reach the cluster head. In a multi-hop network nodes use multi-hop communication to forward sensing data to reach the cluster head. In both cases, the cluster heads use single-hop to reach the BS.

1.4.2 Mobility in Wireless Sensor Networks

Recent research [12][13] shows that significant energy saving can be achieved in wireless sensor networks by using mobile devices capable of carrying data mechanically. In this approach, a small number of mobile devices roam about sensing fields and collect data from sensors. As a result, significant network

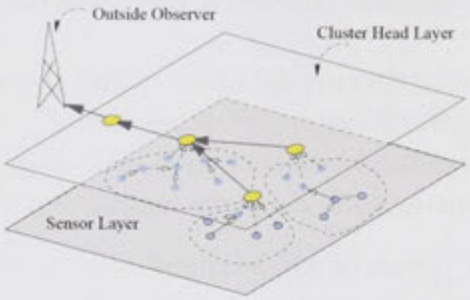


Figure 1.2: A two-layer hierarchical sensor network [2].

energy saving can be achieved by reducing or completely avoiding costly multi-hop wireless transmissions.

Mobility in WSNs can be achieved by using vehicles or people carrying sensors. The energy consumption of mobile devices is less constrained as they can replenish their energy supplies because of their mobility. However, the primary disadvantage of this approach is increased latency. For instance, the typical speed of several practical mobile device systems is approximately $0.1 - 1$ m/s [38]. Thus, it takes more than 16 min for a mobile device to take a tour of length 1 Km to gather sensing data, which may not meet the delay requirements of some data-intensive applications.

In some applications like disaster management it is more efficient to use vehicles since the environmental conditions are harsh. The mobile device could also be carried using aerial and remotely piloted vehicles [39]. For instance, a number of unmanned aerial vehicles (UAVs) such as helicopters can co-operate with ground sensor nodes for data gathering in mission-critical application. Generally, the mobility of BS can be classified into three types according to the mobility pattern of the entity upon which the BS is mounted as follows:

1. Random Mobility: This can be achieved for example when the BS is mounted on humans and animals [40], in this case the probability of the BS collecting all sensing data is low since the BS opportunistically visits sensor nodes.

2. Predictable Mobility: In this case the BS is mounted on an entity that moves on a fixed track or path that cannot control its direction or speed, but which moves at a regular time, for example a BS mounted on a bus or train. Thus, the sensor nodes can predicate when the BS may move around to send their data [41][42].

3. Controlled Mobility: When the BS is mounted on a robot or UAV plane, then the direction and speed of the BS can be controlled. Several algorithms are

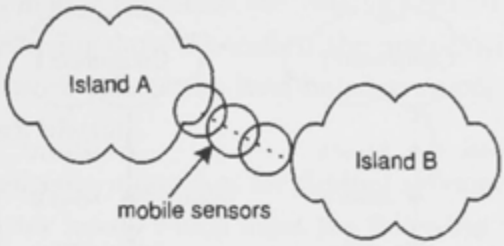


Figure 1.3: Mobile sensors (MRs) are used to provide connection between disconnected network [3].

proposed to find the trajectory of the BS in order to achieve some requirement such as maximise network lifetime and data-gathering delay using single and multi-hop relays.

Most literature considers the mobility of WSNs to be predictable or controlled [2, 41, 43, 44, 45, 4] to achieve network performance requirements (some literature considers random mobility to improve network lifetime), which differs from mobile ad hoc networks in which nodes are assumed to move arbitrarily which degrades the network performance by link failures.

In this research we classify mobile entities used in WSNs into two categories: mobile BS and MR. Mobile BS has unlimited energy sources with high buffering and processing capabilities. MRs may be similar to static sensors (limited transmission range and storage) with movement capability or have higher storage capability to buffer and carry sensing data to be sent to the BS.

In general, mobility in WSNs is used to:

- Improve network coverage: Sensor nodes are usually randomly deployed in the sensing field by scattering from aircraft or by robots [46] which cannot be guaranteed to cover the whole area. Coverage requires that each location in the sensing field be monitored by sensors. An MR relocates its position in order to cover the required sensing field.
- Improve network connectivity: MRs relocate to provide a connection path between several non-connected subnetworks [3][47] as shown in Fig. 1.3.
- Carrying data from isolated sensors: MRs with store-carry-forward capability can travel between isolated sensors to collect sensing data and then forward it to a BS [48, 47, 49, 50, 51, 52].
- Improve network lifetime: Mobile BS and MRs can be used to improve network lifetime. Where MRs can move to a new location to help bottleneck

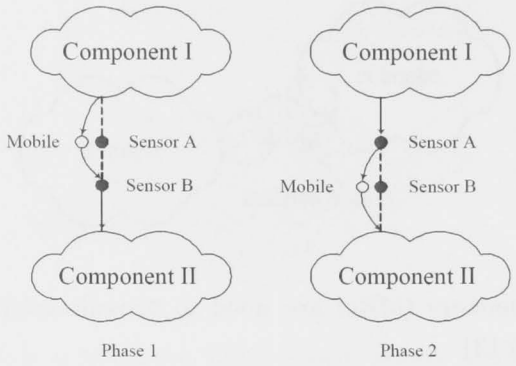


Figure 1.4: Using mobile sensors (MRs) to extend the lifetime of the bottleneck nodes [4].

sensors by inheriting sensing, transmission and receiving responsibilities. For example, Wang et al. [4] assume the network shown in Fig. 1.4. The whole network is composed of two components that are connected via sensors A and B. Thus, these two sensors are the bottleneck nodes since they have to forward all network traffic between the two components. An MR can inherit the responsibility of sensor A and B at some time and thus the network lifetime is improved. A mobile BS and MRs can roam a sensing field (connected/disconnected) and gather data from sensor nodes through a short transmission range. The energy consumption of each sensor node is then reduced, since fewer relays are needed for the sensor node to relay its data packet to the BS. In order to maximize network lifetime, mobile BS and MR can collect data using single-hop communication, however, that increases data-gathering delay since the mobile BS and MRs have to visit the transmission range of each sensor node. On the other hand, using multi-hop communication for data gathering decreases data-gathering delay and decreases network lifetime. Therefore, path planning of a mobile BS and MRs is a critical issue in maximization of network lifetime and to meeting data-gathering delay requirements.

1.5 Contributions

Our contribution in this research can be divided into two parts:

- We consider data gathering in a mobile BS environment, subject to a specified tour delay-time constraint on the mobile BS, by adopting a clustering-based approach. To reduce the energy consumption of a cluster head to forward

sensing data, the mobile BS roams the sensing field and visits only the cluster heads to gather sensing data. Therefore, the distribution of the cluster heads in the entire network affects the load balance among the sensor nodes and hence the network lifetime.

We propose a heuristic algorithm for finding a trajectory of the mobile BS consisting of cluster heads which meet the following criteria: (*i*) the energy consumption among the sensor nodes within any cluster is balanced in order to prolong network lifetime; and (*ii*) the total traversal time of the mobile BS on the trajectory is bounded by a given value. The proposed algorithm significantly increases the network lifetime.

We then conduct a detailed analysis to the proposed algorithm. We analytically study the upper and lower bounds on the number of clusters such that there is no packet lost due to moving too fast through a cluster or interference between cluster heads. Statistical methods are used to determine the probability of finding cluster heads and of losing packets as the BS moves from one cluster to another.

- We propose a routing scheme to provide interaction between MRs and BS, where MRs are used for data gathering from sensor nodes that are moving at relatively lower speed than MRs. An analytical model is needed to study the effect of mobility of MRs on routing performance as the mobility inevitably incurs additional overhead in data-communication protocols, where that overhead can potentially offset the benefit brought by mobility.

  We consider a multipath routing extension of DSR, where separate routes via each neighbour are stored in the cache. In this case the number of neighbouring nodes, and hence paths, effects the network overhead. We develop an analytical model, verified by simulation, for the network as a function of the MR speed, distance to the BS, and node density. The results reveal that the proposed routing scheme can significantly reduce routing overhead, however, the number of cached routes stored should be limited to six to prevent overhead blowout when the MR moves quickly.

1.6 Thesis Overview

In Chapter 2 we present the background and related work. We introduce the techniques used to reduce the energy consumption and evaluate the network lifetime in static sensor networks. We highlight the use of mobility to reduce energy consumption and prolong network lifetime. We define the mobile entities in WSN

as mobile BS and MRs. We survey the related literature that consider finding moving trajectory for a mobile BS and the interaction among MRs.

In Chapter 3 we introduce the idea of reducing the energy consumption of sensor nodes using mobile BS for data gathering. We consider data gathering in a mobile BS environment, subject to a specified tour delay time constraint on the mobile BS, by adopting a clustering-based approach.

In Chapter 4 we provide analysis of the proposed algorithm. We show the benefit and limitation when a mobile BS is used for data gathering. We analytically study the upper and lower bounds on the number of clusters such that there is no packet lost due to moving of BS too fast through a cluster or interference between cluster heads. We use statistical methods to determine the probability of finding cluster heads and of losing packets as the BS moves from one cluster to another.

In Chapter 5 we propose an analytical model to estimate the hop count between source-destination pairs when the network nodes are uniformly distributed in the sensing field. This model is used for calculating the expected overhead in Chapter 4. To calculate the number of hops in the path, we determine a distribution describing the remaining distance from next-hop node to destination. We calculate the probability of selecting each of the neighbour nodes. The expected number of hops needed to cover the remaining distance is calculated by obtaining the expected progress towards the destination. The hop count model is verified by simulation.

In Chapter 6 we propose a routing scheme for MRs interaction. Multipath routing is consider to provide redundant paths to destination. The link residual time between MR and its neighbours is analysed. We explore how the number of paths and spread of neighbour nodes used by the source MR to reach the destination affects the network overhead. The speed of MRs is evaluated in order to minimize the network overhead due to mobility. An analytical model is developed and verified by simulation.

Finally, in Chapter 7, we present an overview of the results presented and suggestions for future work.

Chapter

2

Background

Energy consumption is a crucial consideration for sensor networks and their applications as sensor nodes are commonly battery-driven. Once sensor nodes are deployed, it is challenging and sometimes even impossible to change batteries. Hence, the network lifetime becomes a critical concern in the design of WSNs.

The lifetime of a sensor network can be defined in different ways, according to the effect of losing sensor nodes on the functionality of the whole network, which depends on the sensor network application [53]. For instance, network lifetime could be defined as the time until the first node depletes its battery. On the other hand, a network could be considered to be alive as long as a given percentage of the sensors has enough energy to operate. In this case, the network lifetime is defined as the time for which a given percentage of the region is covered by live sensors. In this research we define network lifetime as the time the first node fails, which is the definition most frequently found in the literature.

Data gathering in sensor networks can be divided into two types according to the specific needs of the applications, these are time-driven and event-driven. In a time-driven scenario all sensors send data periodically to the sink. As opposed to this, in the event-driven case sensors start communicating with the sink only if sensing an event, i.e., a situation that is worth reporting according to the requirements of the application. In this research we address the time-driven scenario, and provide energy-efficient solutions for homogeneous networks, with sensors having constant and equal amounts of data to send in all parts of the sensing field. In the following section, we list some techniques that are used to reduce energy consumption in ad hoc and sensor networks.

2.1 Techniques to Reduce Energy Consumption

Nodes in wireless ad hoc and sensor networks consume energy in sensing, processing, transmission and reception. Communication functions expend most of the node energy. In the following we present different techniques to reduce energy consumption of ad hoc and sensor nodes related to communication.

2.1.1 Energy-efficient Routing Schemes

Sensed data in ad hoc and sensor networks are usually sent to its destination using multiple-hop communication. The level of transmission power is assumed to be adjusted to the minimum level required to ensure the intended receiver is within the transmission range. The selection of the next hop in the routing path effects the energy consumption of nodes, since the power level will be adjusted depending on the choice of the next hop node [54]. The aim of this technique is to maximize network lifetime by selecting:

- a path to minimize the total energy consumed to reach the destination, which minimizes the energy consumption per unit flow; and/or
- the path with nodes with the highest residual energy.

The network lifetime can also be prolonged by using a multipath routing protocol with a view to providing load balancing among network nodes. A multipath routing protocol can choose to divert traffic through alternative paths to ease the burden of the congested link. Moreover, multipath routing can provide fault tolerance by having redundant information routed to the destination via alternative paths. This reduces the probability that communication is disrupted in the case of link failure which minimizes the control overhead required to discover a new route.

The selection of paths is one of the challenging issues in multipath routing that effect the network performance and has been studied extensively in the literature. The most commonly used criterion is the disjointness of paths, which classifies the paths in terms of shared resources as follows [55]:

- Node-disjoint: There are no common nodes in the paths except the source and destination nodes.
- Link-disjoint: There are common intermediate nodes but no common links.

- Partially disjoint: There are common links and intermediate nodes among paths.

Selecting paths with minimum common nodes and links reduces the probability that a link-failure effects multiple paths. However, selecting paths with disjoint-nodes is difficult in some situations and the discovered paths may have more hops than the shortest hop path, which is not energy efficient [11]. Therefore, in Chapter 6 we propose a routing scheme to find multipath routes between the source and destination. The routes are selected via different neighbour nodes for the source and destination in order to reduce the probability of path failure.

2.1.2 Data Aggregation

Data aggregation is a common technique used in sensor networks. In most WSN applications, sensor nodes are used to detect environmental events and send sensing data to a BS. In event-driven applications, it is required to detect a particular environmental phenomenon (for example to detect when a temperature exceeds 60 degrees at specified time intervals). During that time interval all sensors send their data packets towards the BS. Thus, there are a large number of data packets needing to be sent to the BS, with some of them redundant since some sensor nodes are used to detect the same environment events.

The base idea in data aggregation techniques is to reduce packet complexity at the intermediate nodes and minimize energy consumption of the sensor nodes by taking advantage of correlations among sensing data [56]. Correlation refers to the data redundancy between two sensor packets due to the overlap of sensing activities. The nature of correlation differs with the type of applications considered [57].

Clustering can be used to reduce the amount of data required to be sent to a BS. The sensing field is divided into small clusters, a cluster head being responsible for aggregating and relaying to the BS the data gathered from the sensors of its clusters [58][59]. One of the most cited clustering approaches is LEACH (Low-Energy Adaptive Clustering Hierarchy), a self-organising, adaptive protocol where the nodes organise themselves into clusters, with one node acting as the cluster head. The cluster head aggregates the sensing data and then transmits the compressed data to the BS [5]. This can achieve a reduction in energy consumption, as data aggregation is much cheaper than communication.

Once the cluster-head has all the data from the nodes in its cluster, the cluster-head node aggregates the data and then transmits the compressed data to the BS. In Chapter 3 we consider data gathering by adopting a cluster-based approach. Sensor nodes send their data to the cluster heads. The cluster heads

forward sensing data to the BS without data aggregation. Thus, the cluster heads consume more energy than another node and are the bottlenecks of the network. In the future work, we will consider the scenario that the cluster heads aggregate sensing data in order to reduce the amount of data required to be sent to the BS.

2.1.3 Topology Control

The performance of the network can be impacted in a major way by network topology. The topology of an ad hoc network is the set of links that are available to routing protocols. These links are determined by a number of factors, such as the geographic positions of the nodes, transmission powers assigned to transceivers and signal interference. Dense networks may induce high interference, which, in turn, reduces the effective network capacity due to limited spatial reuse and may cause unnecessarily high energy consumption. In contrast, a sparse network is vulnerable to network partitioning due to node or link failures.

Topology control generally refers to selecting an appropriate transmission power for each node in order to reduce energy consumption and signal interference without impeding performance. All nodes within the actual (large) transmission range are called *physical neighbours*. Typically, each node selects a few *logical neighbours* from its physical neighbours within the normal transmission range, and the (smaller) actual transmission range of each node is set to be the distance to its farthest logical neighbour [60]. That is, nodes keep only some of their physical neighbours as logical neighbours for routing and topology maintenance. Topology control is designed to satisfy global constraints, such as network connectivity, reduced channel contention and other reliability and throughput related measures in addition to reducing the energy consumed by node communication [61, 62, 63].

Topology control problems are usually described as graph problems. The desired effect of topology control is to reduce energy consumption, reduce interference between adjacent nodes, and to increase the effective network capacity. The following are some of the properties topology control tries to achieve:

- k-node-connected: For any integer $k = 1$, the resulting graph remains connected with the removal of any set of $(k-1)$ nodes. In this dissertation, the property connected means 1-node-connected, which is equivalent to saying that there is at least one path between any pair of nodes. Generally, k-connected means k-node-connected, which is equivalent to saying that there are at least k node-disjoint paths between any pair of nodes.
- k-edge-connected: For any integer $k = 1$, the resulting graph remains connected with the removal of any set of $(k-1)$ edges.

- Degree bounded: It is also desirable that the node degree in the constructed topology is bounded from above by a small constant. A small node degree could reduce the contention and interference between neighbour nodes, and also may help to mitigate the well known hidden and exposed terminal problems. In addition, a structure with a small node degree will improve the overall network throughout.

- Neighbours θ-separated: The directions between any two logical neighbours of any node are separated by at least an angle θ, which reduces the signal interference. It also can be used to reduce the receiving power cost when a directional antenna is used. In Chapter 6 we use the neighbours θ-separated to reduce network overhead by decreasing the probability of link failure cased by node mobility.

- Bounded-diameter: The diameter of a graph cannot exceed a prespecified constant. The graph diameter is the largest number of nodes that must be traversed in order to travel from one node to another when paths that backtrack, detour, or loop are excluded from consideration. Hence, the graph diameter is equal to the longest shortest path between any pair of nodes in the graph.

- Planar: A network topology is also preferred to be planar (no two edges crossing each other in the graph) to enable some localized routing algorithms to work correctly and efficiently without using a routing table. Each intermediate node can decide which logical neighbouring node to forward the packet to using only local information and the position of the source and the destination [61].

Besides the above described techniques, there are several other techniques built on the monitored regions being usually covered redundantly by sensors; in such a case it is essential to provide optimal sleep scheduling solutions so as to minimize redundant reporting while maintaining high coverage ratios. Generally these techniques are based on finding a trade-off between energy savings and the coverage area [64][65].

2.2 Mobility to Reduce Energy Consumption

Mobility in WSNs has been proposed to prolong network lifetime by balancing the energy consumption among sensor nodes [2, 41, 43, 44, 45, 4]. In the following section, we discuss the network lifetime at static BS by studying the routing-energy consumption, then we introduce the use of mobile BS and MRs to improve network

lifetime. We focus on the following: trajectory of mobile BS, effect of BS speed, the consideration of data gathering delay and packet loss and how the MRs interact with each other in order to send sensing data to the BS.

2.2.1 Static WSNs

In [5] the authors demonstrated an analysis of energy consumption for a network with a static BS for two conventional approaches of routing protocols, direct communication with the BS and multi-hop routing.

In the direct communication protocol, each sensor sends its data directly to the BS. If the BS is far away from the nodes, direct communication will require a large amount of transmit power from each node. This will quickly drain the battery of the nodes and reduce the system lifetime. However, all data reception in this protocol occurs at the BS, so if the BS is close to the nodes, or the energy required for sensors to receive and transmit data is large, this may be an acceptable method of communication.

The second approach considered is a minimum-energy routing protocol. There are several power-aware routing protocols discussed in the literature [66, 67, 68, 69]. In these protocols, nodes route data destined ultimately for the BS through intermediate nodes. Thus nodes act as routers for other nodes in addition to sensing the environment.

For this minimum-transmission-energy routing protocol, each data packet must go through a number of low-energy transmits and receives. Depending on the relative costs of the transmit amplifier and the radio electronics, the total energy expended in the system might actually be greater using minimum-transmission-energy routing than direct transmission to the BS as illustrated in [5]. That is, when transmission distance is short and/or the radio electronics' energy use is high, direct transmission is more energy-efficient on a global scale than minimum-transmission energy routing.

The direct communication and minimum-transmission energy routing also effects the energy consumption of each individual node. Fig. 2.1 shows the number of sensor nodes that remain alive as time progresses. It is assumed that after the energy dissipated in a given node reaches a set threshold, the node was dead for the remainder of the simulation. The plot shows that the number of sensor nodes die out more quickly using minimum-transmission energy routing than for direct transmission.

The location of sensor nodes that remain alive and those that are dead is shown in Fig. 2.2. The sensor nodes closest to the BS die out first for

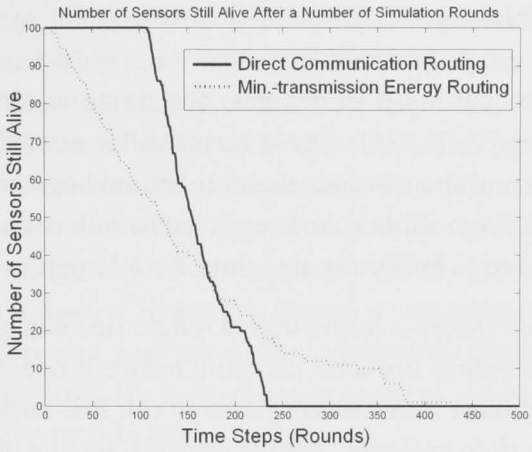
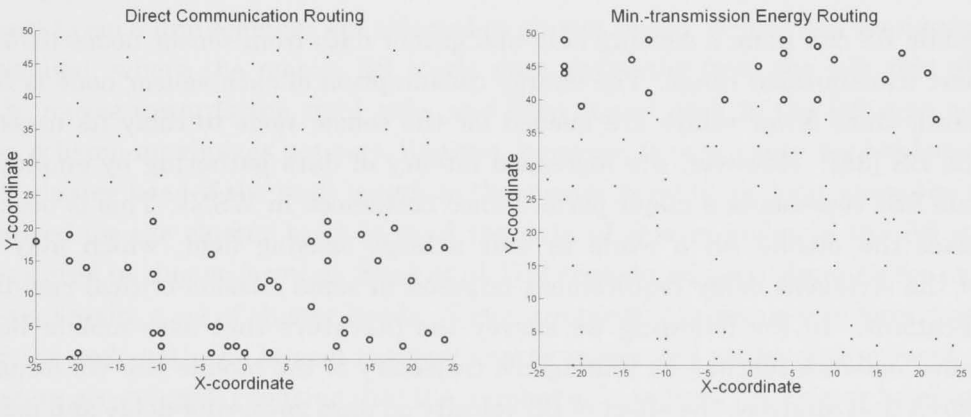


Figure 2.1: Network lifetime using direct communication and minimum-transmission energy routing [5].



(a) (b)

Figure 2.2: Sensors that remain alive are indicated by circles and sensors that are dead are indicated by dots. The BS located at $100\ m$ from the closest sensor node, $x = 0$, $y = -100\ m$. (a) For direct routing, (b) For minimum-transmission energy routing [5].

minimum-transmission energy routing, since the nodes closest to the BS are used to route a large number of data packets to the BS. Thus these nodes die out quickly, causing the energy required to get the remaining data to the BS to increase and more nodes to die. Therefore, the area of the environment is no longer being monitored. On the other hand, the results also show that the sensor nodes farthest from the BS die out first for direct transmission since they have the largest transmission energy.

2.2.2 Mobile WSNs

Different terms have been used to describe the name of mobile entities used in WSNs [41, 44, 70, 71, 72, 73, 74]. These terms differ according to the properties of the mobile entities and the wireless communication between mobile entities and sensor nodes. *Mobile BS* or *Mobile sink* are used to collect data from sensor nodes so that they do not have to buffer sensing data for a long time.

When the network covers a large area, a single BS may not be sufficient, even if it is mobile. *Mobile relays* are data carrying network nodes which collect, carry and forward buffered data from sensor nodes to the BS. Variants of MRs are also referred to as *mobile data collector*, *mobile element*, *mobile observer*, *data ferry* or *data mule* [12]. In this research we will refer to this type as MRs.

2.2.2.1 Mobile Base Station

A mobile BS can roam a sensing field and gather data from sensor nodes through a short transmission range. The energy consumption of each sensor node is then reduced, since fewer relays are needed for the sensor node to relay its message to the BS [38]. However, the increased latency of data gathering by employing mobile BSs represents a major performance bottleneck in WSNs. This is because it takes the mobile BS a while to tour a large sensing field, which may not meet the stringent delay requirement imposed in some mission-critical real-time applications. In the following we survey the literature that uses mobile BS to improve network lifetime by finding the trajectory of the mobile BS. We focus on research that analyses the effect of BS velocity on data gathering-delay and packet loss.

Kansal et al. [75] combine multi-hop forwarding with a mobile BS. They studied performed an experimental evaluation for a small sensor network, assuming that a mobile BS moves back and forth on a straight line (a fixed path). They employed a directed diffusion approach to gathering sensed data from the sensor nodes beyond the transmission range of the mobile BS.

Other literature considered changing the speed of the mobile BS while it traverses the sensing field. Sugihara and Gupta [76, 77, 13] proposed heuristics for finding routing paths for mobile BSs. They assumed that the BS can select the path and change its speed under a predefined acceleration constraint to achieve minimum data-delivery latency and minimize the energy consumption of sensor nodes.

Some existing studies assume that sensor nodes can cache sensing data of

other nodes in order to forward it to the mobile BS when the BS moves around. Gao and Zhang [78] and Xing et al. [38] considered the delay requirement for data gathering assuming that the sensor nodes have an ability to cache sensing data of other nodes. In [78] sensor nodes send their data through multi-hop relays to the nodes (sub-sink) located within the direct-transmission range of the mobile sink. The sub-sink nodes cache data and send it to the mobile sink when it comes within transmission range. In [38] a rendezvous-based data-gathering approach is proposed, in which a subset of nodes are chosen as rendezvous points. The role of these points is to buffer and aggregate data originating from sensor nodes. When the mobile BS arrives within the transmission range of rendezvous points the data will be forwarded to the mobile BS.

Clustering sensor nodes in the sensing field into clusters and collecting data from cluster heads is proposed in [6, 79, 80]. Ma and Yang [6] proposed a heuristic for finding routing paths for the mobile BS (SenCar), which assumes that the moving path of the mobile BS consists of a series of line segments. Sensor nodes closest to each line segment are selected as cluster heads. A specified configuration is applied, where the mobile BS starts data gathering from the left side of the path, moves towards the right side, and then comes back to the left side again. This scheme maximizes network lifetime, however, it may cause packet losses at each cluster head if the path length to the cluster head is too long, since the time required for the cluster head to send the data of cluster nodes to the BS is not considered in cluster forming. Saad at el. [79] considered path planning for mobile BS that visits a set of cluster heads. Nodes are randomly grouped to form clusters and the node with the largest residual energy is selected as the cluster head. The clusters are merged, ensuring that the number of hops from the cluster head to any cluster node is no more than two hops. The cluster head gathers and buffers cluster sensing data. The BS visits the cluster heads to gather data through single-hop communication. However, this work does not consider data gathering delay for a mobile BS. Zhang at el. [80] extended the work in [38] and regarded cluster heads as the rendezvous point. They considered event based data gathering and proposed reduction in data gathering delay by only allowing the mobile BS to visit cluster heads that generate new data.

Luo and Hubaux [43] proposed an analytical model to find a trajectory of the mobile BS for data gathering through multi-hop relays. They showed that the optimal tour of the mobile BS is the perimeter of the sensing field.

Table 2.2 summarizes the related literature in terms of clustering sensor nodes, considers data gathering delay, the effect of BS speed and packet losses due to BS mobility. The final row shows the differences between our algorithm proposed in Chapter 3 and analysed in Chapter 4.

Reference	Clustering	Data gathering delay	Speed effect	Considered packet loss
Kansal et al. [75]	No	No	Yes	Yes, speed control
Ma and Yang [6]	Yes	No	No	No
Saad at el. [79]	Yes	No	No	No
Sugihara and Gupta [76, 77, 13]	No	Schedule to reduce latency	Yes	Yes
Gao and Zhang [78]	No	Yes	No	No
Xing et al.[38]	No	Yes	No	No
Luo and Hubaux [43]	No	No	No	No
Chapter 2 and 3	Yes	Yes	Yes	Yes

Table 2.1: Comparison considered for mobile BS in the literature.

It can be seen that none of the literature considers organizing sensor nodes into clusters for data gathering within a specified delay time. Ma and Yang [6] and Saad at el. [79] only considered clustering sensor nodes and collecting data from cluster heads. However, they did not consider data gathering delay. Gao and Zhang [78] and Xing et al.[38] considered the data gathering delay. However, they assumed that the sensor nodes have the ability to cache sensing data of other nodes. In addition, the reports did not consider the effect of mobile BS speed on the number of clusters and packet loss when the BS moves from one cluster to another.

Kansal et al. [75] proposed a speed control technique to reduce the number of packet losses. However, they assumed the mobile BS moves on a fixed path and uses a direct diffusion approach for data gathering. Sugihara and Gupta [76, 77, 13] considered the BS can change its speed in order to minimized data-delivery latency and minimize the energy consumption of sensor nodes. However, they did not consider organizing sensor nodes into clusters for data gathering. In contrast, we assume sensor nodes are organized into clusters and the cluster heads are visited by the BS within a specified delay time. Clustering in our algorithm is determined by equal-sized geographic regions. Statistical methods

are used to determine the probability of finding cluster heads and of losing packets as the BS moves from one cluster to another. We then examine how the resulting network lifetime varies with node density. We assume the average speed of the mobile BS is constant, as changing the speed of the mobile BS leads to significantly higher manufacturing costs and power consumption.

In Chapter 4 we analyse the algorithm proposed in Chapter 3, in particular we calculate the number of clusters such that there is no packet lost as the BS moves between clusters. No discussion of the number of clusters appears in Ma and Yang [6] or Saad at el. in [79], however, this topic has been addressed for static WSNs.

Calculating the optimal number of clusters in WSNs has been addressed in literature using different methods [81, 36, 82]. However this literature did not consider a mobile BS. Guo at el. [81] proposed a cluster scheme to address the limitation of control channel bandwidth and sensing delay problems. They demonstrated that the overhead and delay are reduced as the number of clusters is decreased. The optimal number of clusters is selected to balance the tradeoff between the communication overhead and required sensing performance. Mhatre and Rosenberg [36] determine the number of clusters when the sensor nodes communicate with the cluster head using single and multi-hop communication. The cluster head aggregates the received data and sends it directly to the BS. Wang at el. [82] proposed a cross-layer analytical model to determine the optimal number of clusters that minimized average energy consumption.

2.2.2.2 Mobile Relays

MRs are mobile sensor nodes that roam the sensing field and collect sensing data using single or multi-hop communication. Sensing data are buffered and carried in order to forward to the BS. MRs can interact with each other, and may have the same communication range and sensing ability as static sensor nodes, MRs inherit the sensing and relaying responsibilities of the bottleneck nodes they are close to, allowing bottleneck nodes to sleep for some time to save energy. MRs can also be requested to move to other positions to improve connectivity among the sensor nodes. In sparse ad hoc and sensor networks, MRs can provide packet delivery where they can buffer and carry packets across network partitions to be forwarded to other nodes. In the following we survey some of the literature that uses MRs to achieve network performance improvements.

Several studies have considered the MR and routing path for data gathering using single-hop communication. Ma and Yang [83] proposed heuristics for prolonging network lifetime for data gathering from all sensors by finding a short

routing path for one or more MRs. The MR traverses the network, collecting sensor data directly within single-hop nodes.

Somasundara et al. [71] studied the routing path of MRs that traverse the sensing field at constant speed, where the sensor nodes operate at different sampling rates. An algorithm is proposed to find a path that minimizes the buffer overflow at each sensor node based on the constraint of buffer and data generation rates in each node.

An MR with a constrained path is introduced by Chakrabarti at el. in [41] and [84]. The MR moves on a fixed path and collects sensing data using single-hop communication. A sensor node predicts MR arrival time, wakes up, and starts data transfer with the MR. Thus a reduction in power can be achieved. They analyse the requirement for successful communication between the relay and sensor nodes which is based on sensor-transmission range, velocity of the relay and time the sensor node has to wait before starting to send its data.

Other studies consider multi-hop communication for data gathering. Jea et al. [85] assume MRs move on a straight line, and propose a load-balancing algorithm assuming mobile-sensor nodes fully cover the entire field of the network.

Using the MR as cluster head is proposed in [86, 2, 87]. Ma et al. [86] propose a three-tier architecture: sensors, mobile sink and BS tiers. The sensor nodes in the sensor tier are organised in a cluster with MRs as the cluster heads. MRs roam the cluster nodes and buffer the sensing data. In order to send data to a BS, MRs can communicate with each other and with the BS through short or long communication radios. They study the effect of relay velocity on message-delivery delay and outage probability when the MRs move randomly. Ma and Yang in [2] proposed an algorithm to prolong network lifetime by finding a path for the cluster heads in the sensing field. The cluster heads are equipped with powerful transceivers and batteries and can communicate directly with each other and with the BS. Banerjee et al. [87] propose the MRs have the capability to buffer sensing data and send it to the the sink when the sink is within its transmission range. Sensor nodes form clusters and select MRs as the corresponding cluster heads. Each cluster head roams among its cluster nodes and collects sensing data using single-hop or multi-hop paths. The collected data is buffered and is then forwarded to the sink using direct communication. The cluster head maintains a connected path to the sink all the time while it is moving. They propose a cluster head connectivity algorithm that restricts the movement of the cluster heads in order to ensure that the cluster heads can communicate with each other.

Other studies make use of the residual energy of sensor nodes in order to prolong network lifetime. Marta and Cardei [88] consider multiple mobile BSs

deployed for data gathering. The mobile BSs move along a predetermined path past sensors with high residual energy, such that the BSs remain interconnected at all times.

Initial deployment of sensor nodes in the sensing field can be changed using MRs where MRs move to other positions after deployment to improve connectivity among sensor nodes and minimize energy consumption by reducing distance between hops in the path between source-destination pairs. The MRs move to form a straight path between source-destination pairs as the most energy efficient multipath can be achieved as a straight path between the source and destination [89]. The energy cost of relay mobility in addition to the energy required for communication for single and multiple data flow is considered in [90, 91, 73] where they address the scenario that the energy replenishment of MR cannot always be possible due to the constraints of the physical environment. MRs are used to relay data and use the same transmission range as sensor nodes. Goldenberg et al. [90] and Tang and McKinley [91] propose mobility control algorithms that each relay node moves to the midpoint of its neighbour coverage. The communication energy is minimized by reducing the distance between hops in the path between the source and destination. In contrast El-Moukaddem at el. [73] show that the optimal position of an MR depends on the amount of data to be sent in addition to the initial position of sensor nodes. This work increases the reliability of the path between the source and destination by improving the connectivity between communicating neighbours which can avoid the cost of control packet overhead incurred in requesting a new route.

In addition, mobility is used to provide packet delivery in partition sensor and ad hoc networks since nodes are often scattered in sensing fields by aircraft or robots. These randomly scattered nodes may cause sensors to be partitioned into disconnected subnetworks. An MR is used to buffer and carry packets across a network partition, and forward packets to other nodes when they meet [92].

MRs with short-range wireless communications and buffering capability in partition sensor networks are proposed by Shah et al. in [40]. There is no communication among the MRs and the random walk model is used to describe the mobility of MRs. The MR collects and buffers sensing data from sensor nodes when sensors are within its transmission range. Several wired access points are set up to provide network connectivity with the sink. The collected data is sent to an access point when the MR moves the access point.

Zhao and Ammar in [92] proposed algorithms to find the route of multiple MRs in a partition ad hoc network under the constraint of traffic demand and data delivery delay. MRs move at a constant speed and use the same radios as nodes.

They studied two ways MRs interact with each other, either directly or via nodes. The MRs need to synchronize their movements to exchange data directly, while nodes are required to have enough storage and energy for buffering and relaying data among MRs. The same authors in [72] proposed an algorithm to find the route of an MR under the constraint of data-delivery delay when the network nodes can adapt their trajectory to meet the relay and transmit or receive packets. Bin Tariq at el. in [91] assume the network nodes are move arbitrarily so that it is difficult to determine when the MR can contact mobile nodes.

Reference	Path	Muli-hop forwarding	Speed effect	Communication range of MRs	Send data to the BS	Mobility type
Ma and Yang [83]	Varies	No	No	Long	Direct/ via others MRs	Controlled
Jea et al. [85]	Fixed	Yes	Yes	Long	Via other MRs	Controlled
Marta and Cardei [88]	Fixed and varied	Yes	Move and stop	Long	Via other MRs	Controlled
Somasundara et al. [71]	Varies	No	No	Long	Unspecified	Controlled
Ma et al. [86]	Unspecified	Yes	No	Long	Via other MRs	Predictable & randomly
Chakrabarti et al. [41][84]	Fixed	No	Yes	Long	Unspecified	Predictable
Ma and Yang [2]	Varied	Yes	No	Long	Via other MRs	Controlled
Banerjee et al. [87]	Varied	Yes	No	Long	Via other MRs	Controlled
Shah et al. [40]	Unspecified	Yes	No	Short	Via access point	Randomly
Zhao and Ammar [92]	Varied	Yes	No	Same as node	Via other MRs/nodes	Controlled
Chapter 6	Varied	Yes	Yes	Same as node	Via other nodes	Controlled or randomly

Table 2.2: Comparison of using MRs.

Table 2.2 summarizes the related literature in terms of path planning, using multi-hop forwarding for data gathering, considering the effect of BS speed, communication range of the MR, the method used for the MRs to interact with each other and the mobility type of MRs. Most of the literature assumes that the MRs change their routing path dynamically, use multi-hop for data gathering and did

not consider the effect of the speed of MRs. In addition they assumed the MRs used long communication radios so that they can communicate with each other and with the BS. MRs with short-range wireless communication is proposed in [40] where access points are used to provide network connectivity with the sink. [92] assumes the MRs use the same communication radios as sensor nodes. Controlled direction of MRs for data gathering are assumed by most literature.

The final row in Table 2.2 shows how our scenario proposed in Chapter 6 differs from the literature surveyed here. In contrast to the other work on MRs, we assume the MRs to move freely through the sensing field and instead focus on a routing scheme to pass information between MRs. We assume MRs use a short transmission range or the same as sensor nodes so that sensor nodes are used to provide a routing path via multi-hop communication. We consider the effect of MR speed on network overhead due to path failure among MRs when they roam a sensing field for data gathering.

In the next chapter we introduce the idea of reducing the energy consumption of sensor nodes using mobile BS for data gathering. We consider data gathering in a mobile BS environment, subject to a specified tour delay time constraint on the mobile BS, by adopting a clustering-based approach.

Chapter

3

Mobile Base Station Tour Algorithm

3.1 Introduction

Energy consumption of sensor nodes in WSNs is a crucial factor as sensor nodes are small devices that are typically powered by small batteries. Mobile BSs are used to reduce the power consumption of sensor nodes [44, 90, 43] by roaming the sensing field and gathering data from sensor nodes through a short transmission range. The energy consumption of each sensor node is then reduced, since fewer relays are needed for the sensor node to relay its message to the BS [38].

In Chapter 2 we demonstrated the limitation in network lifetime for static WSNs. In Section 2.2.2.1 we introduced the idea of reducing the energy consumption of sensor nodes using mobile BS for data gathering. In this chapter, we consider data gathering in a mobile BS environment, subject to a specified tour delay time constraint on the mobile BS, by adopting a clustering-based approach. To reduce the energy consumption of a cluster head to forward sensing data, the mobile BS roams the sensing field and visits only the cluster heads to gather sensing data. Therefore, the distribution of the cluster heads in the entire network affects the load balance among the sensor nodes and hence the network lifetime.

We propose a heuristic algorithm for finding a trajectory of the mobile BS consisting of cluster heads which meet the following criteria: (*i*) the energy consumption among the sensor nodes within any cluster is balanced in order to prolong network lifetime; and (*ii*) the total traversal time of the mobile BS on the trajectory is bounded by a given value. The proposed algorithm significantly increases the network lifetime.

3.2 Preliminaries

Consider a data gathering application, such as environmental monitoring, in which all the sensing data must be delivered to the BS within a specified delay time. Our optimization objective is to prolong the network lifetime by minimizing the energy consumption of sensor nodes using a mobile BS.

3.2.1 Network Model

We make the following assumptions about the network:

1. The transmission range, r, of each sensor node is fixed and identical.
2. All sensor nodes have identical initial energy, and the mobile BS replenishes its energy periodically so that there is no energy concern with the mobile BS.
3. The speed of relaying a data packet by sensors is much faster than the moving speed, V_m, of a mobile BS ($T_P << T_C$), where T_P is the time required to transmit a data packet to the BS and T_C is the time taken by the mobile BS to traverse the transmission range of the cluster head. The total delays, D, in data gathering can be mapped into the maximum length, L, of a BS tour ($D = L/V_m$).
4. There is sufficient time to establish communication and send one or more data packets during the time the BS takes to travel across the transmission range of a sensor node ($T_{rq} + T_P << 2r/V_m$) where T_{rq} is the time from when the mobile BS enters the transmission range of the cluster head to when it receives the first sensing data.
5. Sensor nodes are densely deployed in the sensing region (average node degree ≥ 8). Accordingly, the number of hops in a path is approximately proportional to the distance between the nodes (as demonstrated in Chapter 5).
6. The storage of a sensor node is limited, so that it cannot buffer a large volume of data.
7. Sensor nodes and the mobile BS are assumed to know their own physical locations via a Global Positioning System (GPS) or a location service in the network.
8. The BS moves with constant velocity and must change direction smoothly (for example to avoid wheel slippage or mechanical damage during the mobile BS navigation).

3.2.2 Problem Definition

Given a network with a mobile BS, assuming that the length of a BS tour is bounded by L, and its speed is V_m, the problem is to find a tour for the mobile BS such that the network lifetime is maximized. In addition, to minimize the end-to-end delay by reducing the number of hops between the source node and the BS.

3.2.3 Clustering Solution

We propose that sensor nodes be organized into clusters such that all the cluster heads can be visited by the mobile BS, where the length of the BS tour is no longer than L. The location of the cluster head in its cluster is an essential factor in balancing the energy consumption of the cluster-sensor nodes. In addition, cluster head location affects the length of the BS tour and end-to-end delay between the source sensor and the BS. The challenge in this problem is to find the optimal locations of cluster heads by jointly considering the BS tour, network lifetime and end-to-end delay.

Our technique aims for an equal number of sensors in each cluster in order to achieve load balance among the cluster heads, since each cluster head has to forward the data packets to the mobile BS. In addition, to select cluster heads that are the closest to the centre of cluster in order to reduce the number of hops and hence end-to-end delay between source and the BS. Other researchers use quite different criteria for forming clusters [77][6]. We contend that our technique balances energy consumption and data gathering time among the cluster heads.

Assume that n sensor nodes are uniformly distributed in a sensing field with area A. For a given number of clusters, K, the sensing field is divided into equal sub-areas and the area of each such sub-area is A/K. All the nodes located in the same sub-area form one cluster. If n_{K_i} is the number of sensor nodes in cluster K_i, $1 < K_i \leq K$, then the expected number of sensor nodes in this cluster is $E(n_{K_i}) = nA_{K_i}/A$, where A_{K_i} is the area of cluster K_i.

3.3 Algorithm

To determine the BS route, we first determine clusters, then identify a virtual cluster head for each cluster, and finally identify sensor nodes which are real cluster heads. The BS route is a smooth trajectory passing over each real cluster head. To determine the best possible locations for cluster heads in order to

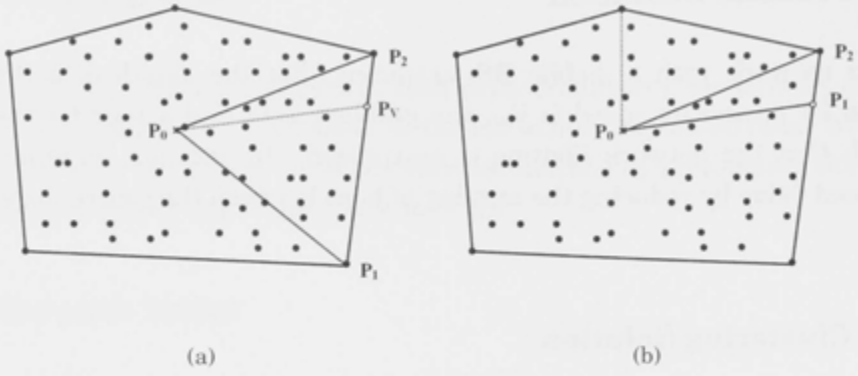


Figure 3.1: An example of clustering procedure, P_0 is the centroid location of sensing field, P_1 and P_2 are the locations of two boundary sensor nodes, $P_V \in \overline{P_1P_2}$, and A_K is the cluster area. (a) Area $(P_0, P_1, P_2) \geq A_K$. (b) Area $(P_0, P_1, P_2) < A_K$.

maximize the network lifetime, two issues must be considered. The first is how to cluster the sensor nodes in the entire network such that (i) all the cluster heads can be visited by a mobile BS; and (ii) the length of the BS tour is no greater than the given tour length, L. The second is the selection of cluster heads to balance the energy consumption among sensor nodes within each cluster. The cluster-heads are the bottlenecks of energy consumption, since they have to forward the sensing data of sensor nodes within them to the mobile BS. Thus, to maximize the network lifetime, energy consumption among the cluster heads needs to be balanced, which can be achieved by partitioning the entire sensor field into equal sub-areas. To organize the sensor nodes into clusters, each sensor node is assigned to the sub-area in which it is located. Thus, energy consumption among the cluster heads will be balanced since the sensor nodes are uniformly deployed in the sensing field. Next, it is required to find the cluster heads by jointly considering the energy consumption of sensor nodes in the entire cluster and the length of the BS tour, L. To balance energy consumption among sensor nodes, it is important to select the cluster head such that each sensor node in a cluster is within a certain number of hops from its cluster head. Accordingly, the sensor nodes near to the cluster centre become the candidates for the cluster head if the length of the BS tour is no greater than L. In subsections 3.3.1-3.3.3 we propose a detailed algorithm for the problem.

3.3.1 Clustering

We aim for an approximately circular tour around the sensing field. To accommodate this, the clusters are arranged radially. The intention of clustering is to

divide the sensing field into equal sub-areas by radial lines from the centre of the field, therefore the sensor nodes on the boundary of the sensing field need to be determined. Graham's scanning algorithm is applied to find a set of the boundary sensor nodes, B, for the convex polygon of the sensing field. In this polygon, each sensor node is either on the boundary or inside of the polygon. The area of the polygon can be calculated using the locations of boundary sensor nodes (X_i, Y_i), $1 \leq i \leq n_B$, where n_B is the number of boundary sensor nodes and (X_i, Y_i) is the location of a boundary sensor node. The area A_P and the centroid location (X_0, Y_0) of the polygon can be found as follows [93]:

$$A_P = \frac{1}{2}\sum_{i=1}^{n_B}(X_i Y_{i+1} - X_{i+1}Y_i), \tag{3.1}$$

$$X_0 = \frac{1}{6A_P}\sum_{i=1}^{n_B}(X_i + X_{i+1})(X_i Y_{i+1} - X_{i+1}Y_i), \tag{3.2}$$

$$Y_0 = \frac{1}{6A_P}\sum_{i=1}^{n_B}(Y_i + Y_{i+1})(X_i Y_{i+1} - X_{i+1}Y_i), \tag{3.3}$$

where $X_{n_B+1} = X_1$ and $Y_{n_B+1} = Y_1$.

For a given number of clusters, K, the sensing field is divided into equal sub-areas (cluster area) and the area of each such sub-area is $A_K = A_P/K$. The sub-areas are determined by selecting an arbitrary sensor node on the boundary, B_1, as the starting point, P_1, then selecting the second sensor node, P_2, on the boundary in anti-clockwise order. The area bounded by P_1, P_2 and the centroid P_0 is calculated, using (3.1). If this area is greater than A_K, this means that the required area must be bounded by P_0, P_1 and an intermediate point (a virtual sensor node) P_V that lies on the $\overline{P_1P_2}$ line, as shown in Fig. 3.1(a). Making use of A_K and the locations of P_0, P_1 and P_2, the location of P_V is calculated as follows:

$$Y_V = \frac{Y_1 - Y_0}{X_1 - X_0}X_0 + Y_0 - \frac{Y_1 - Y_0}{X_1 - X_0}X_0, \tag{3.4}$$

$$X_V = \frac{1}{Y_0}(X_0 Y_V - 2A_K). \tag{3.5}$$

If the calculated sub-area is less than the cluster area as shown in Fig. 3.1(b), a new sensor node on the boundary of the sensing field next to P_2 needs to be added, and the area has to be re-calculated. The details of the clustering procedure are shown in Procedure 1. The procedure finds the set of boundary cluster sensor nodes, BK.

Procedure 1 The Clustering Procedure.
Input: K, A_P, B, P_0
Output: Set of boundary cluster sensor nodes BK

Select an arbitrary sensor node P_1 from B;
$A_K = A_P/K$;
$A = A_K$ store cluster area;
for *each cluster* i, $1 \leq i \leq K$ **do**
  $Flag = False$;
  Put P_0 and P_1 as the boundary sensor nodes into BK_i;
  while *not Flag* **do**
    Select sensor node P_2 next to P_1 in anti-clockwise order;
    if $Area(P_0, P_1, P_2) \geq A$ **then**
      Calculate a virtual sensor node P_V using (3.4) and (3.5);
      Add P_V as a boundary sensor node into BK_i;
      $P_1 = P_V$;
      $A = A_K$;
      $Flag = True$;
    else
      $A = A - Area(P_0, P_1, P_2)$;
      Add sensor node P_2 into BK_i;
      $P_1 = P_2$;
    end
  end
end

3.3.2 The Calculation of the Location of Virtual Cluster Heads

The next step is to determine the candidates for cluster heads. According to assumption 3.2.1.4, the number of hops from a sensor node to its cluster head is approximately proportional to its distance to the cluster head. Therefore, to minimize the number of hops between a cluster head and its sensor nodes, this distance must be minimized. Thus, the centre of a cluster area is a desirable location for the cluster head, because it balances the energy consumption among the sensor nodes in the cluster. Using (3.2) and (3.3), the potential location of the centre of the cluster areas, PC_i, can be found, with A_P replaced by the cluster area, A_K, and X_i and Y_i restricted to the cluster boundary.

The length, L_K, of a tour through each of the PC_i can then be obtained. If L_K is less than L, these PC_i will be considered as a virtual cluster head, VCH, since the mobile BS can move back and forth along the route; otherwise the tour length

must be reduced through relocating PC_i towards the centre of the sensing field. To achieve a load balance among cluster sensor nodes, the same amount of relocating is employed to each cluster. The reduction rate is the ratio of L to L_K. Simple geometric equations are derived to find VCHs to satisfy the required tour-length constraint. Procedure 2 describes the procedure for calculation of the location of VCHs and Fig. 3.2 illustrates the concept of relocating VCHs.

Procedure 2 Calculation of The Location of Virtual Cluster Heads.
Input: L, K, A_K, BK, $P_0(X_0, Y_0)$
Output: Set of virtual cluster heads $VCH(XV, YV)$

for *each cluster* i, $1 \leq i \leq K$ **do**
  Calculate cluster centroid point $PC_i(XC, YC)$ using (3.2) and (3.3) with BK_i parameter and $A_P = A_K$;
end
L_K = Length of the tour connecting PC points;
if $L_K \leq L$ **then**
  $VCH_i = PC_i$, $1 \leq i \leq K$;
else
  $\alpha = L/L_K$;
  for *each cluster* i, $1 \leq i \leq K$ **do**
    $l_i = \alpha\sqrt{(XC_i - X_0)^2 + (YC_i - Y_0)^2}$;
    $\theta_i = \tan^{-1}(YC_i/XC_i)$;
    $XV_i = l_i \cos(\theta_i)$;
    $YV_i = l_i \sin(\theta_i)$;
    $L_K = L$;
  end
end

3.3.3 Finding Real Cluster-head Sensor Nodes

So far the locations of VCHs have been calculated. In order to ensure the tour length meets the delay requirement, sensor nodes close to each VCH are candidates for the corresponding real cluster head. In order to determine which candidate node will be chosen in each case, we consider the length of a sector of the BS tour that connects the candidate cluster head with the previous and next VCHs (termed the 'Real Segment' (RS)). We also refer to a sector of the BS tour that connects the current VCH with the previous and next VCHs as the 'Virtual Segment' (VS). The real and virtual segments are both used as references in order to decide whether the candidate cluster head will increase or decrease the total BS tour length. An

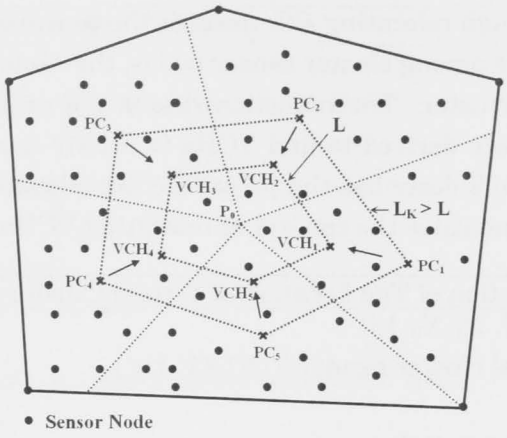


Figure 3.2: An illustrative example of virtual cluster heads calculation, for $K = 5$ clusters. PC_i and VCH_i, $1 \leq i \leq K$ are the locations of clusters' area centre points and virtual cluster heads, respectively.

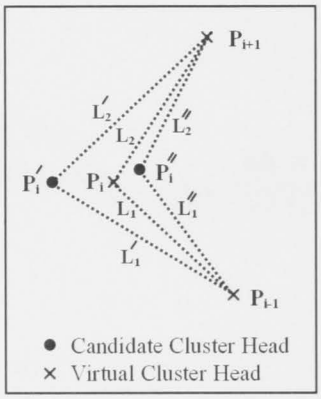


Figure 3.3: An example of real and virtual segments. Candidate cluster heads and VCHs are denoted by black circles and crosses, respectively. P_i refers to a point, which may be a VCH or a sensor node. The virtual segment for P_i is $L_i = L_1 + L_2$, while the real segments for $P_i^{'}$ and $P_i^{''}$ are $L_i^{'} = L_1^{'} + L_2^{'}$ and $L_i^{''} = L_1^{''} + L_2^{''}$, respectively. $L_i < L_i^{'}$ and $L_i > L_i^{''}$.

example of real and virtual segments is shown in Fig. 3.3. To find a set of real cluster heads to form a tour such that the tour length is no greater than L, two sensor nodes close to each VCH need to be found, where one sensor node increases the length of the tour while the other decreases it. To identify two such candidate cluster heads, the sensor nodes in each cluster are sorted in increasing order of the distance to the corresponding VCH. Then, for the first sorted sensor node, the real segment is calculated and compared with the virtual segment. The sensor node is assumed to increase the length of the BS tour if the real segment is greater than

the corresponding virtual segment, otherwise, it decreases the length of the tour.

Procedure 3 Finding Two Candidate Cluster Heads for Each Virtual Cluster Head- Finding Real Cluster Head Procedure.

Input: VCH, K, L
Output: Set of real cluster heads RCH

Initiate status flag to unfinish for K clusters, $FK_i = False$, $1 \leq i \leq K$;
Initiate real cluster head $RCH_i = VCH_i$, $1 \leq i \leq K$;
for *each cluster* i *in* K **do**
  Set a flag for each candidate cluster head, $FI = False$; $FD = False$;
  VS_i=length of virtual segment connecting VCH_i with VCH_{i-1} and VCH_{i+1};
  SN=Set of all cluster sensor nodes in i-th cluster ordered according to increasing distance from VCH_i;
  while *not FI or not FD* **do**
    if *SN is not empty* **then**
      Pick a node n from SN;
      RS_i=length of real segment connecting n with VCH_{i-1} and VCH_{i+1};
      if $RS_i \geq VS_i$ **then**
        if *FI=False* **then**
          Add node n into NI_i set;
          Store $\Delta LI_i = RS_i - VS_i$;
          $FI = True$;
        end
      else
        if *FD=False* **then**
          Add node n into ND_i set;
          Store $\Delta LD_i = RS_i - VS_i$;
          $FD = True$;
        end
      end
    else
      $RCH_i = NI_i$;
      $FK_i = True$;
    end
  end
end

The checking continues until the two candidate cluster heads closest to the corresponding VCH are found. If a candidate cluster head that decreases the length of the tour cannot be found, the candidate cluster head that increases the tour

length is selected as a real cluster head. The details of finding two candidate cluster heads for each VCH are shown in Procedure 3.

The selection of real cluster heads proceeds in two phases. In Phase I, the closest candidate cluster head to the corresponding VCH is selected as a real cluster head, if the length of its real segment is less than the corresponding virtual segment. Thus, all not-chosen candidate cluster heads closest to their VCHs have a real segment greater than the corresponding virtual segment. What follows is the search for a real cluster head that has not yet been selected, so that the length of the BS tour is no greater than L. To find such a cluster head, the BS tour is calculated using the candidate cluster head, the real cluster heads which have been selected so far, and the VCHs of the remaining clusters. The candidate cluster heads closest to the VCH of each of the remaining clusters are sorted in increasing order according to the difference in length between real and virtual segments $(VS - RS)$. For the first candidate cluster head, the total length of the BS tour is calculated. If it is less than L, the candidate cluster head is selected as the real cluster head. For the next candidate cluster head, the total BS tour length is calculated, having taken into account the selection of the previous one. Phase I will terminate once it finds a candidate cluster head with the corresponding length of the BS tour greater than L. Ultimately, all the remaining candidate cluster heads that are closest to the VCH have RS greater than VS. The details are shown in Phase I.

Phase II initially assumes that the remaining candidate cluster heads with RS less than VS are selected as the real cluster heads; therefore the length of the BS tour is less than L. However, it is possible to find a real cluster head closest to the VCH, with RS greater than VS and where the total BS tour is less than L. To find such a cluster head, the candidate cluster heads that have not been selected from Phase I are sorted in increasing order of distance from their corresponding VCHs. For the first candidate cluster head, the total length of the BS tour is calculated, using the real cluster head selected from Phase I, in addition to the initial selected cluster head from Phase II after changing the initial selected cluster head with the corresponding candidate cluster head. If the length of the BS tour is less than L, the candidate cluster head is finalized and selected as the real cluster head. For the next candidate cluster head, the length of the BS tour is calculated by taking into account the selection of the previous one, and so on. Ultimately, all the initially selected cluster heads are finalized and selected as the real cluster heads. Phase II and Fig. 3.4 show the details of Phase II and an example, respectively.

Phase I Finding Real Cluster Heads Procedure.

for *each unfinished cluster* i *in* FK **do**
  if $\Delta LI_i > |\ \Delta LD_i\ |$ **then**
    $RCH_i = ND_i$;
    $FK_i = True$;
  end
end
SN=Set of increasing sort order of candidate cluster heads in NI_i for the unfinished clusters according to length ΔLI_i;
for *each unfinished cluster* i *in* FK **do**
  Pick a candidate cluster head CH from SN;
  L_i=Length of the BS segments connecting real cluster heads in RCH by using CH instead of RCH_i;
  if $L_i < L$ **then**
    $RCH_i = CH$;
    $FK_i = True$;
  else
    exit loop;
  end
end

Phase II Finding Real Cluster Heads Procedure.

for *each unfinished cluster* i *in* FK **do**
  $RCH_i = ND_i$
end
for *each unfinished cluster* i *in* FK **do**
  SN=Set of increasing sort order of candidate cluster heads in NI_i according to its distance to the corresponding VCH_i;
  Pick a candidate cluster head CH from SN;
  L_i=Length of the BS tour segments connecting real cluster heads in RCH using CH instead of RCH_i;
  if $L_i < L$ **then**
    $RCH_i = CH$;
    $FK_i = True$;
  else
    exit loop;
  end
end
Set $FK = True$ for all unfinished clusters;

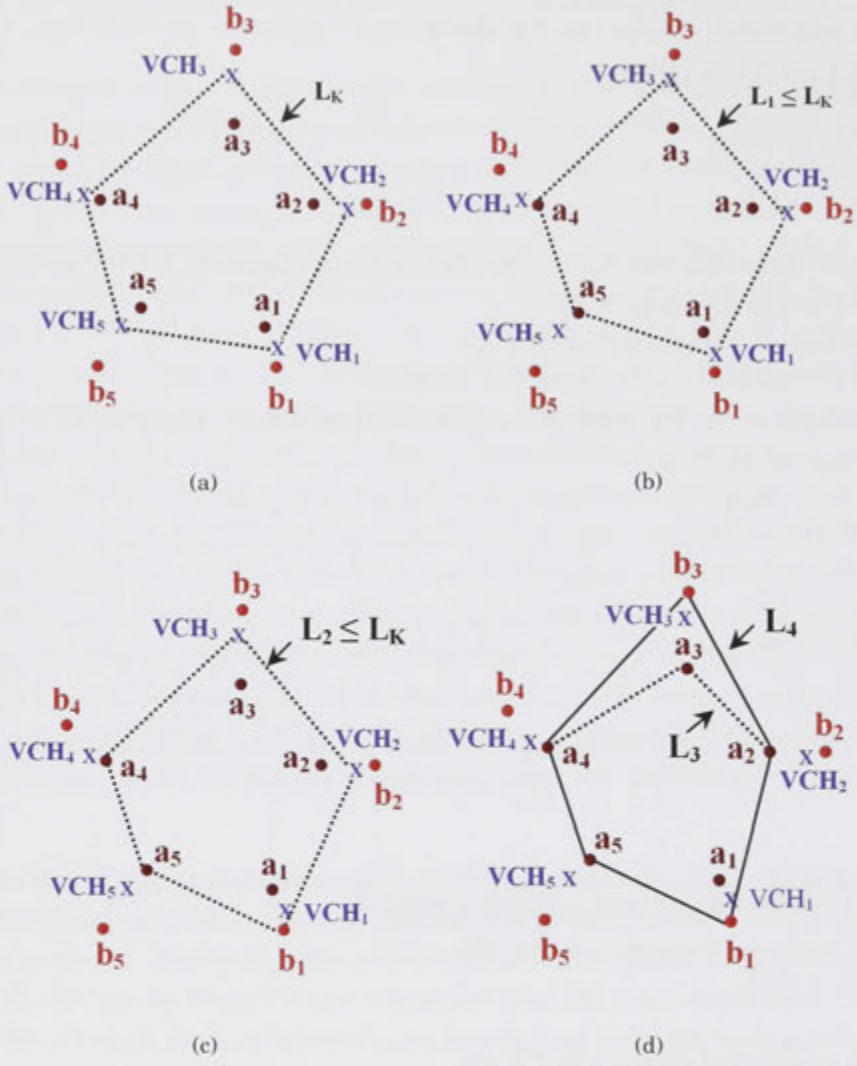


Figure 3.4: An execution example of finding real cluster heads, $K = 5$ clusters. (a) Finding two candidate cluster heads for each cluster, one with $RS_i \geq VS_i$ and the other with $RS_i < VS_i$, $1 \leq i \leq K$. (b) After the execution of Phase I, sensor nodes a_4 and a_5 are selected as real cluster heads since they are closest to VCH_4, VCH_5 and $RS_4 < VS_4$, $RS_5 < VS_5$, respectively. (c) Then, sensor node b_1 is selected as real cluster head since it is closest to VCH_1 and $L_2 \leq L_K$. (d) After the execution of Phase II, sensor nodes a_2 and a_3 are initially selected as real cluster heads, then a_3 is changed with b_3 as real cluster head since it is closest to VCH_3 and $L_4 \leq L_K$, b_2 is not selected as a real cluster head since the corresponding BS tour length is greater than L_K.

3.4 Conclusion

In this chapter we dealt with the problem of data gathering in the mobile BS environment subject to the constraint that the sensing data needs to be gathered within a specified delay. We presented a clustering-based heuristic algorithm for finding a trajectory of the mobile BS to balance the energy consumption among sensor nodes. The proposed algorithm allows the BS to visit all cluster heads within a specified delay. Our technique aims for an equal number of sensors in each cluster in order to achieve load balance among the cluster heads, since each cluster head has to forward the data packets to the mobile BS. In Chapter 4 we analyse the algorithm to determine effective choices for the parameters to achieve maximum network lifetime.

Chapter

4

Analysis of Mobile BS Tour Algorithm

4.1 Introduction

In Chapter 3 we proposed a heuristic algorithm for finding the trajectory of a mobile BS consisting of cluster heads, which meet, the following criteria: (*i*) the energy consumption among the sensor nodes within any cluster is balanced in order to prolong network lifetime; and (*ii*) the total traversal time of the mobile BS on the trajectory is bounded by a given value.

In this chapter we provide an analysis of the proposed algorithm. We analytically study the upper and lower bounds on the number of clusters such that there is no packet lost due to moving too fast through a cluster or interference between cluster heads. Statistical methods are used to determine the probability of finding cluster heads and of losing packets as the BS moves from one cluster to another. Table 4.1 shows the main symbols used in this chapter.

4.2 Choosing the Number of Clusters

The algorithm for finding the tour of a mobile BS employs the number of clusters, K, as a system parameter. In this section we aim to analytically study the upper and lower bounds of the number of clusters needed. The minimum number of clusters, K_{min}, is determined by the maximum number of nodes that can be in a cluster before packets begin to be lost, and the maximum number of clusters, K_{max}, arises from the requirement that the transmission regions of the cluster heads do not intersect.

Symbol	Description
D	Data gathering delay: time available for a mobile BS tour
r	Transmission range of a sensor node
L	Maximum allowable length of the mobile BS tour, $L = DV_m$
L_m	Actual length of the mobile BS tour
L_K	Length of the mobile BS tour connecting cluster centroids
V_m, V_{max}	Average and Maximum speed of the mobile BS, respectively
T_{rq}	Packets request time, time between data request to receiving the first data packet
T_P	Packet time: average time required to transmit a data packet to the BS
T_C	Contact time
T_R	Residual contact time
T_{pks}	Total time required to send data packets of the cluster nodes to the BS ($T_{pks} = n_K T_P$)
R	Radius of the network field
n	Network nodes
K	Number of the clusters
n_K	Number of nodes in a cluster
n_s	Number of nodes the cluster head can successfully send their data packets to the BS
l	Number of packet losses

Table 4.1: Definitions of the main symbols used throughout this chapter.

4.2.1 Minimum Number of Clusters

The cluster head transmits sensing data when the mobile BS is within its transmission range. The transmission time available to the cluster head is determined by the speed of the BS, thus there is a maximum number of packets that can be sent in that time. The network must have at least K_{min} clusters to ensure that the transmission load for each cluster head is not too high.

When the mobile BS reaches the transmission range of a cluster head, it advertises its presence by periodically broadcasting a special packet called a

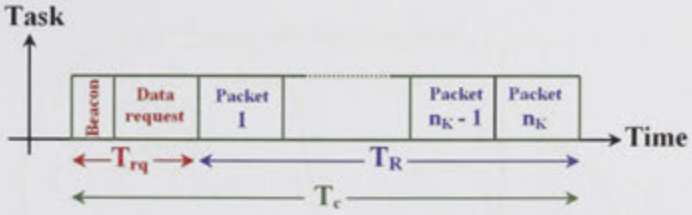


Figure 4.1: The mobile BS data gathering scenario.

beacon. A cluster head, upon receiving the beacon, broadcasts the beacon packet, requesting that the cluster nodes send their data to the cluster head. Let the time from when the mobile BS enters the transmission range of the cluster head to when it receives the first sensing data be T_{rq} and the time taken by the mobile BS to traverse the transmission range of the cluster head be T_C. Since the mobile BS visits each cluster head, $T_C = 2r/V_m$. We assume that only one packet can be transmitted from the cluster head at a time. Thus, the residual time available for gathering cluster data is $T_R = T_C - T_{rq}$, as shown in Fig. 4.1.

Let T_P be the average time required for the cluster head to collect and send a data packet to the mobile BS. Assume there are n_K sensor nodes in a cluster, then $T_{pks} = n_K T_P$ is the time required to collect the sensing data from that cluster. If $T_R \geq T_{pks}$, no packet loss is incurred in data gathering. However, if $T_R < T_{pks}$, then the residual time is not enough to collect all of the data packets in the cluster. The number of packets that can be successfully transmitted is $n_s = \lfloor T_R/T_P \rfloor$, where $\lfloor x \rfloor$ is the largest integer less than x. In summary, the number of packets lost is given by,

$$P_{loss} = \begin{cases} 0 & T_R \geq T_{pks}, \\ n_K - n_s & T_R < T_{pks}. \end{cases} \tag{4.1}$$

The minimum number of clusters necessary can be found by allowing $T_R = T_{pks}$. If all clusters have the same number of nodes, we would have $n_K = \lceil n/K \rceil$, where $\lceil x \rceil$ is the largest integer greater than x. Substituting into $T_R = T_C - T_{rq}$, we find

$$K_{min} = \left\lceil \frac{nT_P V_m}{2r - V_m T_{rq}} \right\rceil. \tag{4.2}$$

However, nodes are independent and identically uniformly distributed, the number of sensor nodes in each cluster can be modeled by a binomial distribution $n_K \sim B(n, 1/K)$. The probability function of the time required to collect cluster packets is given by

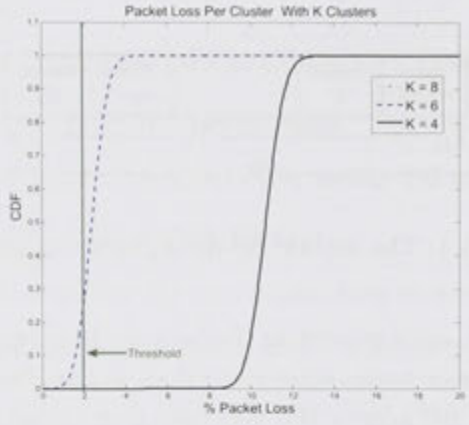


Figure 4.2: The effect of the number of clusters on the CDF of the percentage of number of packet losses, from (4.5) when $r = 100\ m$, $V_m = 2\ m/s$, $n = 3500$, $T_{rq} = 10\ ms$ and $T_P = 200\ ms$, where m meter, s second and ms millimeter. For these settings, the approximate minimum number of clusters, K_{min}, is eight from (4.2).

$$f_{T_{pks}}(t = n_K T_P) = \binom{n}{n_K} \left(\frac{1}{K}\right)^{n_K} \left(1 - \frac{1}{K}\right)^{n-n_K}, \tag{4.3}$$

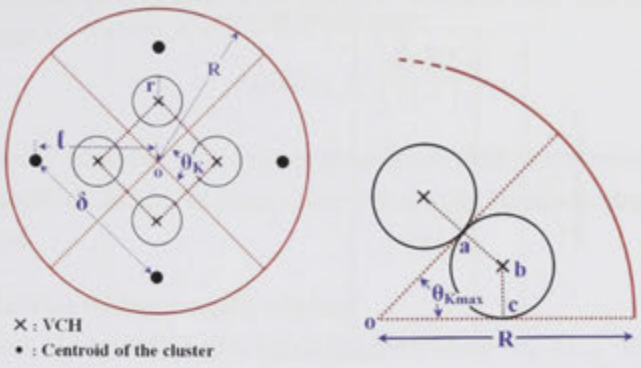
where $t = 0, T_P, 2T_P, ..., nT_P$. The probability that l packets are lost is

$$f_L(l) = \begin{cases} \sum_{t=0}^{n_s T_P} f_{T_{pks}}(t) & l = 0, \\ \binom{n}{l+n_s} (\frac{1}{K})^{l+n_s} (1 - \frac{1}{K})^{n-(l+n_s)} & l = 1, 2, ..., n - n_s. \end{cases} \tag{4.4}$$

The cumulative distribution function (CDF) is given by

$$F_L(l) = \sum_{i=0}^{l} f_L(i). \tag{4.5}$$

Using (4.5), we can study the effect of the number of clusters on packet loss as shown in Fig. 4.2, where the CDF of the percentage of numbers of packet losses is plotted for different numbers of clusters. The network parameters are configured such that the approximate K_{min} is eight. The result shows that the probability of achieving any given threshold level of packet loss increases with the number of clusters. For instance, the probability of achieving packet loss less than 2% is 0, 0.26 and 1 for the number of clusters 4, 6 and 8, respectively.



(a) (b)

Figure 4.3: The maximum number of clusters. (a) Transmission range of VCHs do not overlap. (b) Probability of finding real cluster head is proportional to the ratio of the area oabc to the sensing field.

4.2.2 Maximum Number of Clusters

We have shown that packet loss decreases with the increasing number of clusters due to decrease in the forwarding load for each cluster heads. However, increasing the number of clusters decreases the distance between cluster heads of adjacent clusters, so that eventually the transmission range of cluster heads overlaps, which decreases the effective contact time because transmissions by each cluster head interfere with the others when the BS is in the region where their transmission ranges overlap. Therefore we require that the distance between cluster heads is at least $2r$.

We now investigate the effect of increasing the number of clusters on the length of the BS tour and the probability of finding a real cluster head.

First, assume that the VCH is located at the centroid of the cluster indicated by a bullet in Fig. 4.3(a). Assume the sensing field is a circle, with radius $R > 2r$ (as illustrated in Fig. 4.3(a)) and let θ_K represent the angle between the boundaries of the cluster area. Then the distance between the VCH and the centre of the sensor field is

$$\ell = \frac{2RK\sin(\pi/K)}{3\pi}, \tag{4.6}$$

where $\theta_K = 2\pi/K$ is the internal angle of the cluster, as indicated in Fig. 4.3(a). Then, the length of tour segment between adjacent VCHs is

$$\delta = 2\ell\sin(\pi/K), \tag{4.7}$$

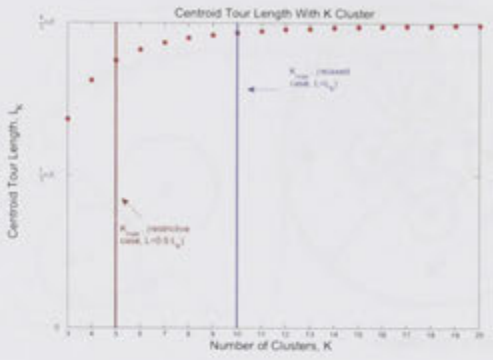


Figure 4.4: The effect of the number of clusters on the BS tour when $r = 1$ and $R = 5r$. K_{max} is calculated from (4.9).

and the length of the BS tour connecting cluster centroids is $L_K = K\delta$. The centroid approaches the perimeter as the cluster becomes narrower, so the tour length increases with the number of clusters to reach approximately 66.7% of the perimeter of the sensing field as shown in Fig. 4.4. This result differs from [43], which shows the optimal tour of the mobile BS is the perimeter of the sensing field because of the use of a different data collection scheme. In [43] the network sensor nodes send their data directly to the BS, however, in our work, data is sent to cluster heads which then forward it to the BS.

Now, let us consider the effect of delay requirements on the tour length, L_m. We consider two cases as follows:

Case I: Relaxed delay requirement

Assume the delay constraint $L > L_K$. In this case VCHs do not need to shrink in from the centroid, thus $L_m = L_K$. The distance between VCHs is L_m/K, so the transmission ranges of VCHs do not overlap when $L_m/K \geq 2r$. The maximum number of clusters can be obtained when the time the BS spends in each cluster is T_C, that is, when the length of the tour segment within each cluster is equal to $2r$ as shown in Fig. 4.3(b), so, $K_{max} = L_m/2r$. Moreover, the internal angle of the clusters in this case is $\theta_{K_{max}} = 2\pi/K_{max}$.

In order to determine K_{max}, Substituting $L_m = L_K$ and $L_K = K_{max}\delta$, we have $\delta = 2r$. Using (4.6) and (4.7), with $\theta_K = \theta_{K_{max}}$, gives

$$2R\sin^2\left(\frac{\pi}{K_{max}}\right) = \frac{3r\pi}{K_{max}}. \tag{4.8}$$

There is no closed-form solution to (4.8), so we use the Taylor series approximation for small values of $\theta_{K_{max}} = 2\pi/K_{max}$. Taking into account

that the number of clusters is an integer. We find that the maximum number of clusters is

$$K_{max} = \left\lfloor \frac{2\pi R}{3r} \right\rfloor \tag{4.9}$$

The maximum number of clusters increases with increasing network radius since the position of the cluster centroid moves towards the perimeter of the sensing field.

Case II: Restrictive delay requirement

Assume that the tour length must be less than L_K ($L_m < L_K$). We choose $L_m = L$. The VCH (indicated by 'x' in the Fig. 4.3(a)) must move in from the centroid, closer to the centre of the sensing field. The maximum number of clusters can be obtained when the length of the tour segment within each cluster is equal to $2r$, so the maximum number of clusters is $K_{max} = \lfloor L/2r \rfloor$.

In summary, the maximum number of clusters is given by,

$$K_{max} = \begin{cases} \lfloor \frac{2\pi R}{3r} \rfloor & L \geq \frac{4\pi R}{3} \quad \text{(Relaxed)}, \\ \lfloor \frac{L}{2r} \rfloor & L < \frac{4\pi R}{3} \quad \text{(Restrictive)}. \end{cases} \tag{4.10}$$

The smallest valid scenario is $R = 2r$. In any reasonable scenario, this would correspond to the relaxed case, so $K_{max} = 4$. However, in this case the transmission ranges of cluster heads overlap when $K = 2$ or $K = 3$. In realistic scenarios, $R >> 2r$, thus $K_{max} > 4$. As R increases, K_{max} increases, up to some point when the restrictive case is triggered. For example, when $R = 5r$, the maximum number of clusters is $K_{max} = 10$ if $L \geq L_K$, but if $L = 0.5L_K$ (restrictive case), then $K_{max} = 5$.

To ensure the tour is no longer than L, each VCH needs to find a corresponding real cluster head. Referring to Fig. 4.3(b), the probability that a single node lies in the region `oabc` is equal to the ratio of the area bounded by `oabc` to the area of sensing field, $A = \pi R^2$, that is

$$p = \frac{RL \sin(\pi/K) \cos(\pi/K)}{3\pi^2 R^2} \tag{4.11}$$

For n network nodes, the probability of finding a real cluster head is

$$Pr = 1 - (1 - p)^n. \tag{4.12}$$

Fig. 4.5 shows that the probability of finding a real cluster head decreases with the number of clusters and with decreasing tour length. Referring to Fig. 4.5, when K_{max} is equal to 10 and 5 for L equal to L_K and $0.5L_K$, respectively, we see that the corresponding probability of finding a real cluster head is approximately one for both cases.

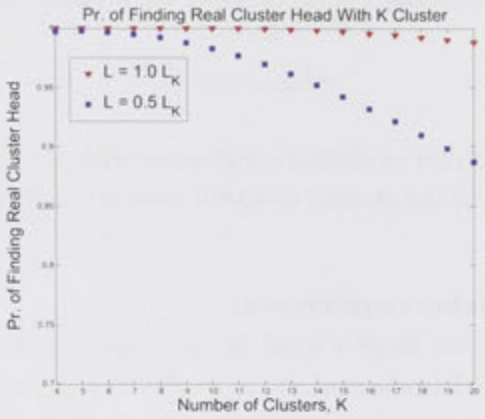


Figure 4.5: The effect of the number of clusters on the probability of finding a real cluster head, $n = 200$, $r = 100\ m$ and $R = 5r$, from (4.12).

Since the real cluster head is close to, but not at the VCH, the transmission ranges of real cluster heads may overlap. The probability that real cluster head transmission ranges overlap increases with the number of clusters and decreasing node density.

4.3 Analysis

In this section we study the effect of data gathering delay, node density and network radius on network lifetime. We also determine the upper bound of network radius as the node density and the velocity of BS varies. Finally, we determine the maximum velocity the BS can move for data gathering such that there is no packet loss for a particular node density. In this section we assume that the number of clusters equals K_{max} from (4.10).

4.3.1 Network Lifetime

All non-cluster-head sensor nodes send their data to their corresponding cluster heads. The cluster heads forward the data to the mobile BS when the BS visits the clusters; thus the cluster heads are the bottlenecks of the network. Assume E_I is sensor node initial energy. Packets are individually transmitted by cluster head with no aggregation. Let E_p denote the average amount of energy required to receive, process and transmit one packet. The amount of energy used by a cluster

head in one cycle is nE_p/K. Thus, the expected network lifetime is

$$E(\text{Lifetime}) \approx \frac{E_I L_m}{E_p V_m n_K}, \tag{4.13}$$

where L_m/V_m is the time required for the BS to complete one cycle. (4.13) represents the maximum network lifetime that can be achieved when n sensor nodes are evenly distributed in K clusters. We can see that increasing the size of the network requires corresponding increase to the number of clusters in order to maintain network lifetime. In the following we study variation to network lifetime with the network radius, assuming that the number of clusters is at its maximum value.

In this research, we use node degree as a measure of node density (rather than the number of nodes in a unit area), since it reflects the number of nodes that can be accessed using the maximum transmission range. Since the network nodes are uniformly distributed in the network field, then the average node degree is

$$d = n\frac{\pi r^2}{A}. \tag{4.14}$$

That is, the average node degree is πr^2 times the node density. Substituting for $n_K = n/K$ into (4.13), then the expected network lifetime is $E(\text{Lifetime}) \approx \frac{E_I L_m K r^2}{E_p V_m d R^2}$. This equation shows that network lifetime increases proportionally to the number of clusters, and decreases proportionally to the node density and the square of the network radius. Let the number of clusters be equal to K_{max}, then using (4.10) the expected network lifetime is

$$E(\text{Lifetime}) = \begin{cases} \frac{8\pi^2 E_I r}{9 E_p V_m d} & \text{(Relaxed).} \\ \frac{E_I L^2 r}{2 E_p V_m R^2 d} & \text{(Restrictive).} \end{cases} \tag{4.15}$$

Eq. (4.15) shows that the network lifetime does not depend on network radius for the relaxed delay requirement. This is because decreases in network lifetime due to increasing node density are canceled by increase in network lifetime due to increasing tour length and number of clusters. However, for the restrictive delay requirement, the network lifetime decreases with decreasing BS tour length and increasing network radius. Therefore, if it is required to achieve a certain level of network lifetime for a large-network scale, then using a single mobile BS may be insufficient, even if we use K_{max} cluster. It may be necessary to consider multiple MRs as proposed in [85].

4.3.2 Packet Loss

Network lifetime decreases as network radius increase, since the number of nodes in each cluster increases for a constant transmission range and node density.

However, the time the cluster head contacts the BS depends on the transmission range of the cluster head and BS speed. Thus, there is an upper bound on the expansion of network size such that there is no packet loss.

There is no packet loss if $T_R \geq T_{pks}$, so the upper bound of the network radius is given by

$$R \leq \begin{cases} \frac{2\pi r(2r - T_{rq}V_m)}{3V_m T_P d} & \text{(Relaxed)}, \\ \sqrt{\frac{Lr(2r - T_{rq}V_m)}{2V_m T_P d}} & \text{(Restrictive)}. \end{cases} \tag{4.16}$$

Where we have used $T_R = 2r/V_m - T_{rq}$ and $T_{pks} = n_K T_P$ and (4.10) and assume the nodes are equally distributed among the clusters. Eq. (4.16) shows that the upper bound of network radius decreases with the increasing of BS velocity and node density in relax delay requirement, while the upper bound of network radius depends on the length of the required BS tour in addition to BS velocity and node density in restrictive delay requirement. For example, for network parameters, $r = 100\ m$, $V_m = 2\ m/s$, $T_{rq} = 10\ ms$, $T_P = 200\ ms$ and $d = 15$. For the relaxed and restrictive ($D = 20$ minute) delay requirements, the approximate upper bound of network radius such that there is no packet loss is $R \leq 6.98\ km$ and $R \leq 2\ km$, respectively, where km kilometer.

More accurately, we can find the probability function and the CDF of number of packet losses at different network radii using (4.4), at relaxed delay requirement, the approximate value of K_{max} appearing in (4.10) when $L = L_K$. The CDF of the number of packet losses is shown in Fig. 4.6, using the same network parameters mentioned. The results show that the probability of high packet loss per cluster increases with network radius. Moreover, the number of clusters needed also increases; therefore, the total packet loss increases with network radius. However, controlling the speed of the BS can help to decrease the number of packet losses. The BS could decrease its speed when it moves within a cluster with a large number of sensors, so that it gets enough time to collect packets of all cluster sensors, and increase its speed in clusters with lower numbers of sensor nodes.

Even controlling the speed of the mobile BS can help to reduce the number of packet losses due to unequal numbers of nodes in the clusters, there are definitely a number of packet losses if $R > 6.98\ km$. In order to achieve no packet losses as the network scale increases, MRs could be used to cooperate for data gathering. The network field could be divided into sub-networks with each MR collecting data from one sub-network and then sending the collected data to the BS.

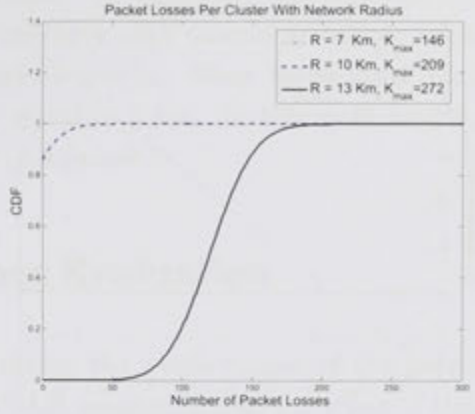


Figure 4.6: The effect of network radius on the CDF of number of packet losses, from (4.4) and (4.5), at relaxed delay requirement, from (4.9). We have $r = 100\ m$, $V_m = 2\ m/s$, $T_{rq} = 10\ ms$, $T_P = 200\ ms$ and $d = 15$, so $n_s = 499$. The approximate upper bound of network radius $R \leq 6.98\ km$ from (4.16).

4.3.3 Maximum Velocity of the Mobile BS

We have shown that the minimum number of clusters is determined by the maximum number of nodes that can be in the cluster before packets begin to be lost, and the maximum number of clusters is determined by the requirement that the transmission regions of the cluster heads do not intersect. When the speed of the mobile BS increases the minimum number of clusters needs to be increased in order to reduce the number of nodes in the cluster so that the BS can collect cluster data within T_R time, as shown in Fig. 4.7. Thus, the maximum speed of mobile BS is determined when the minimum number of clusters increases to reach the maximum number of clusters, $K_{min} = K_{max}$. Using (4.2) and (4.10), the maximum speed of mobile BS is

$$V_{max} = \begin{cases} \frac{4\pi r^2}{2\pi r T_{rq} + 3RdT_P} & \text{(Relaxed)}, \\ \frac{2Lr^2}{LrT_{rq} + 2R^2 dT_P} & \text{(Restrictive)}. \end{cases} \tag{4.17}$$

Eq. (4.17) shows that the maximum speed of the BS decreases with increase to network radius, and decreases even faster in the restrictive delay requirement case, for example, for network parameters, $r = 100\ m$, $R = 1\ km$, $T_{rq} = 10\ ms$, $T_P = 200\ ms$ and $d = 15$, the maximum speed of the mobile BS, $V_{max} = 13.9\ m/s$ and $V_{max} = 10\ m/s$, at relaxed and restrictive ($L = 3\ km$) delay requirements, respectively.

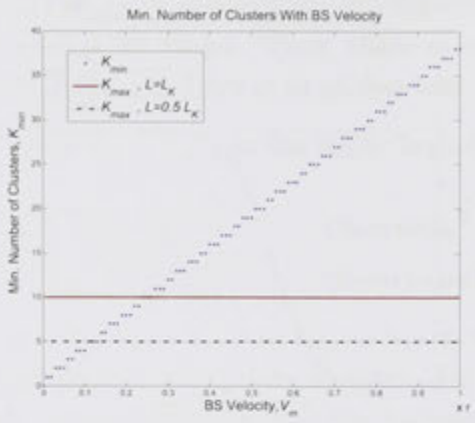


Figure 4.7: The effect of the BS velocity on the approximate minimum number of clusters, from (4.2), when $T_{rq} = 10\ ms$, $T_P = 200\ ms$ and $d = 15$. The approximate maximum number of clusters from (4.10), when $R = 5r$.

4.4 Practical Implications of Analysis

In this section, we consider the use of a mobile BS in practical data gathering applications. The transmission range of sensor nodes varies significantly for different applications. In addition, different types of mobile entities can be used for carrying the BS, for instance, a mobile robot, car, train or UAV plane, so there is a large range for the velocity of the mobile BS.

Assume that a mobile robot that moves at average velocity $2\ m/s$ carries the BS and sensor transmission range equals $100\ m$. When the sensor nodes are clustered with the maximum number of clusters, then there is no packet loss if the network radius is less than $6.98\ km$ for relaxed delay requirements (refer to 4.16 with $T_{rq} = 10\ ms$, $T_P = 200\ ms$ and $d = 15$). In this case the BS tour takes approximately four hours for gathering sensing data since the tour length is approximately $30\ km$, which may be applicable for some applications, but much too slow for others.

Assume that sensing data needs to be collected within 20 minutes. Then the maximum network radius must be decreased to be less than or equal $2\ km$ in order to achieve no packet loss. However, the network radius can expanded further if sensor nodes with higher transmission ranges are used. For example, if the sensors transmission range increase to be $250\ m$, then network radius can be less than or equal $5\ km$.

It is also possible to expand the network while reducing data gathering delay for relax delay requirements by increasing the speed of mobile BS using

for example a UAV plane. In this case, the average speed of BS is 100 km/h (the velocity of some military UAV planes is higher than 200 km/h) and sensor transmission range equals 100 m. Then there is no packet loss if the network radius is less than or equal to 5 km. In this case sensing data can be gathered within approximately 13 minutes.

4.5 Performance Evaluation

In this section we evaluate the performance of the proposed algorithm through simulations with MATLAB, assuming that the effect of the MAC layer is ignored.

We assume that sensor nodes in the network are randomly deployed with uniform distribution in a circular sensing field with radius of $R = 1250\ m$. Each sensor node has a transmission range of $r = 250\ m$ ($R = 5r$) and the initial energy of E_I unit. All data packets have a fixed length and take E_P units of energy per packet. The speed of the mobile BS is assumed to be $V_m = 0.18r\ m/s$. The packet propagation time is $T_P = 200\ ms$ and data request time is $T_{rq} = 100\ ms$. We vary the number of nodes in the network to emulate the change in the node degree. We use the node degree as a metric of node density. For each instance of deployment, the network performance metrics are calculated and the result is the average over 100 instances for each node degree.

4.5.1 Varying Network Scale

We first study the effect of changing the radius of the network field on the network performance. To evaluate the network lifetime of the proposed algorithms, we calculate the maximum network lifetime from (4.13), which assumes all clusters have the same number of nodes. The maximum network lifetime is used as a performance benchmark to see how far away the proposed solutions are from the optimal. We compare our algorithm with the SenCar algorithm proposed in [6]. The moving trajectory of the SenCar (mobile BS) consists of a series of connected line segments and sensors are organized into clusters. The sensor nodes that are nearest to the line segment (which we will refer to as cluster heads) consume more energy than other nodes since they have to forward sensing data to the BS.

The SenCar algorithm assumes a rectangular sensing field and does not implement a closed trajectory for the BS. In order to compare with our algorithm, we assume the SenCar BS moves out across the top half of the circular sensing field and returns across the bottom half.

Fig. 4.8 shows the network lifetime delivered by using our algorithm compared

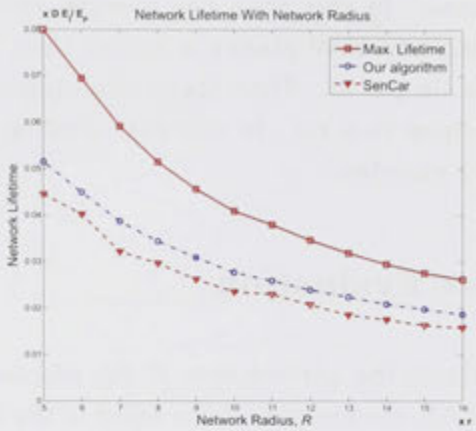


Figure 4.8: Network lifetime as it varies with network radius, for our algorithm, SenCar algorithm [6], and the maximum lifetime for the same numbers of clusters. The maximum network lifetime is calculated from (4.13), which assumes all clusters have the same number of nodes.

with the SenCar algorithm and the maximum lifetime for the same numbers of clusters. It can be seen that under different network radii, the network lifetime for our algorithm is higher than that for the SenCar algorithm. This is because our algorithm balances the network load among the cluster heads by dividing the sensing field into equal areas. In contrast, clusters in the SenCar algorithm have fixed width but varying area, therefore the cluster heads that are close to the centre of the sensing field consume more energy than the others. The results also show that the network lifetime decreases as the network radius increases. This makes sense because the number of packets that need to be forwarded to the BS increases.

To study the effect of the position of cluster heads on the cluster sensor node load balance, the maximum energy consumption for sensor node neighbouring a cluster head is calculated as shown in Fig. 4.9 for our algorithm and SenCar algorithm. The BFS algorithm is used to find the routing tree for each cluster, where the cluster head is the root of the tree.

The result shows that the neighbour nodes in our algorithm consume lower energy than the neighbour nodes in SenCar algorithm. This is because in our algorithm the cluster head and its neighbours are located very close to the centre of cluster area. On the other hand, the location of neighbour nodes in SenCar algorithm depends on the line segments of the BS trajectory. When the location of the cluster head is close to the border of the sensing field, some of its neighbours are responsible for forwarding a larger number of cluster-node packets to the cluster head and hence consume more energy than the others. The result also shows that

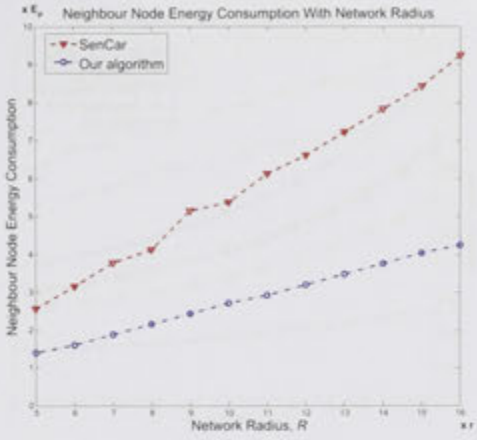


Figure 4.9: The energy consumption for neighbouring sensor nodes of a cluster head for our algorithm and SenCar algorithm [6] as the network radius is varied, for the same numbers of clusters.

the energy consumption increases with the increase in the network radius due to increasing the number of cluster sensor nodes.

4.5.2 Varying Number of Clusters

We then vary the number of clusters and investigate the change in the network lifetime. Fig. 4.10 shows the network lifetime delivered by using the mobile BS compared with the static BS, assuming that the static BS is located at the centroid of the sensing field. The Breadth First Search (BFS) algorithm is used again to find a routing tree rooted at the BS. In the case of a static BS, the BS neighbouring sensor nodes consume more energy than any other sensor nodes in the network since they have to relay the packets received from child sensor nodes to the BS, while in the mobile BS, the cluster heads consume more energy than the other sensor nodes in the network. The network lifetime using the static BS is compared with the maximum and simulated network lifetime of the mobile BS with different numbers of clusters.

Fig. 4.10 shows that the network lifetime decreases as the node degree increases. This makes sense since that increases the number of packets that need to be forwarded to the BS. It can also be seen that as the node degree increases, the difference between the maximum and simulation network lifetime decreases. The reason for this decrease is that the distribution of nodes in each cluster area becomes more even, so the number of packets the cluster heads need to forward become more balanced.

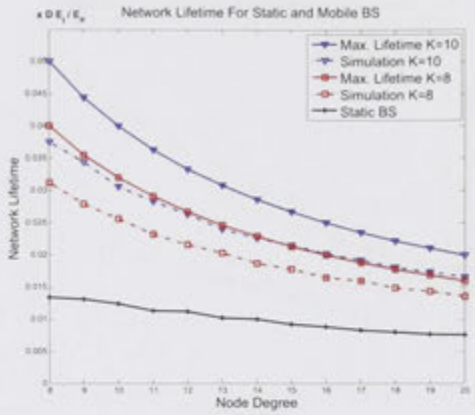


Figure 4.10: Network lifetime as it varies with node degree, for static and mobile BSs with different K. The maximum network lifetime is calculated from (4.13), which assumes all clusters have the same number of nodes.

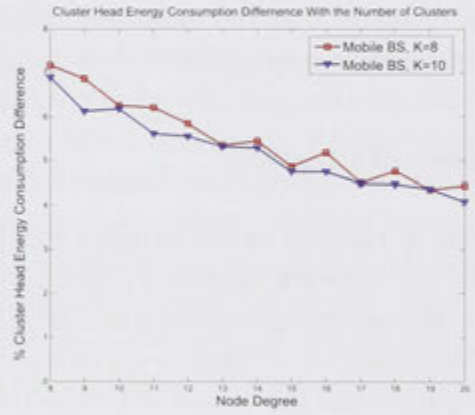


Figure 4.11: The minimum and maximum energy consumption differences among the cluster heads as the node degree is varied.

To evaluate the variation in energy consumption among the cluster heads, we calculate the ratio of difference in cluster head energy consumption to the total energy consumption. The result, shown in Fig. 4.11, shows that the percentage of energy consumption difference decreases with the increasing of node degree. The result also shows that the percentage of cluster head energy consumption difference decreases with the increasing of the number of clusters since that decreases the number of nodes in each cluster.

To study the effect of the number of clusters on the cluster sensor nodes load

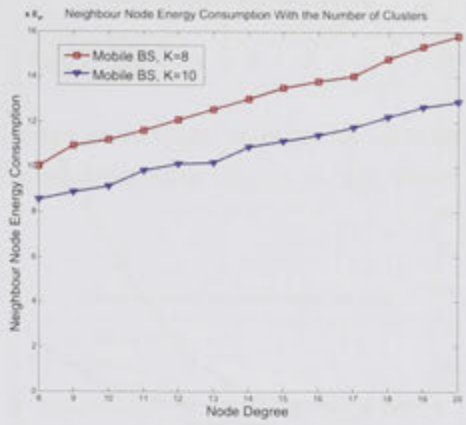


Figure 4.12: The energy consumption for neighbouring sensor nodes of a cluster head as the node degree is varied, for various numbers of clusters.

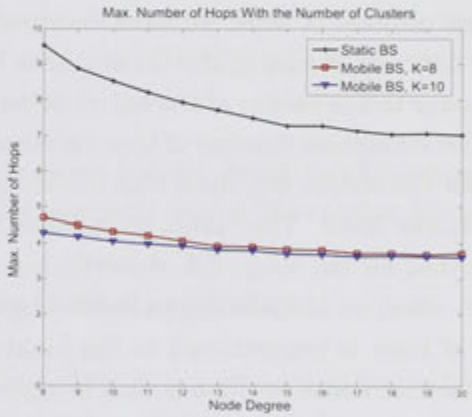


Figure 4.13: The maximum number of hops as it varies with the node degree, for static and mobile BSs.

balance, the energy consumption for neighbouring sensor nodes of a cluster head is calculated as shown in Fig. 4.12. The BFS algorithm is used again to find the routing tree for each cluster, where the cluster head is the root of the tree. The curves in Fig. 4.12 show that the energy consumption increases, with the decrease in the number of clusters due to increasing the number of cluster sensor nodes, and that the neighbouring sensor nodes are responsible for forwarding their data packets to the cluster head. It is also shown that the energy consumption increases, with the increase in the number of network nodes.

Fig. 4.13 illustrates the number of relay hops for the sensing data to reach the

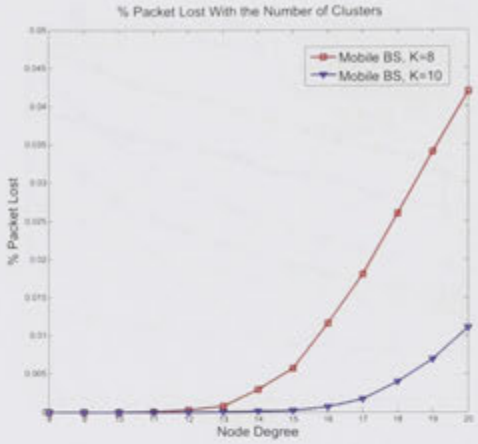


Figure 4.14: The percentage of network packet loss as the number of network sensor nodes are varied, for varies number of clusters.

BS. To find the maximum number of hops, we have to consider the number of hops of sensor nodes located near the border of the cluster area for the mobile BS case, while the sensor nodes near to the border of the entire sensing field are considered for the static BS case. The maximum number of hops increases with the decrease in the number of clusters for the mobile BS, since that increases the distance between a sensor node and its cluster head. The result shows that the maximum number of hops is still less than that for the static BS. Nevertheless, it also shows that the node degree has a small effect on the maximum route length, due to the fact that the maximum number of hops is proportional to the length of the shortest path, since the density of the sensor nodes in the network is high.

The effect of number of clusters on percentage of the network packet loss is shown in Fig. 4.14. The number of packet losses increases with the node degree and decreasing number of clusters, since that increases the number of packets the cluster heads have to forward to the BS within the contact time. Using (4.2) and (4.14), the approximate minimum number of clusters is as shown in Table 4.2. The approximate maximum number of clusters is 10, from (4.10) at relaxed delay requirement. Comparing the number of clusters with the approximate minimum number of clusters, we notice that there is packet loss even when the number of clusters equal or are greater than the minimum number of clusters. This is because the sensor nodes are not equally clustered so that some cluster heads have to forward more data packets to the BS than others.

Node degree	8	9	10	11	12	13	14	15	16	17	18	19	20
K_{min}	4	5	5	5	6	6	7	7	8	8	9	9	10

Table 4.2: The approximate minimum number of clusters as it varies with node degree, from (4.2). The approximate $K_{max} = 10$, from (4.10) at relaxed delay requirement.

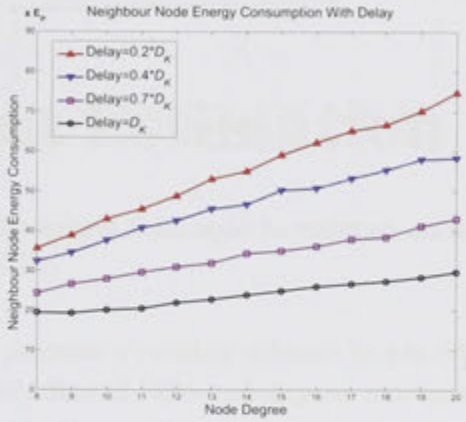


Figure 4.15: The minimum cluster head neighbouring sensor nodes energy consumption as the node degree is varied, for various data gathering delay.

4.5.3 Varying Data Gathering Delay

We finally study the effect of the data gathering delay on the load balance of cluster sensor nodes when the number of clusters is $K = 6$. We define D_K as the time required for the mobile BS to take a tour connecting centroid points of the clusters. The energy consumption for neighbouring sensor nodes of a cluster head is shown in Fig. 4.15. The BFS algorithm is used again to find the routing tree for each cluster. The results show that the energy consumption increases, with the decrease in the end-to-end data gathering delay due to the decreasing length of the BS tour towards the centroid of the sensing field. Therefore, it leads to an increase in the number of sensor nodes, that the neighbouring sensor nodes are responsible for forwarding their data packets to the cluster head. The effect of data gathering delay on the maximum number of relay hops is shown in Fig. 4.16. The maximum number of hops increases with the decrease in data gathering delay for the mobile BS, since that increases the distance between a sensor node and its cluster head. The results show that the maximum number of hops is less than that for the static BS. The results also show that the node degree has a small effect on the maximum route length, since the density of the sensor nodes in the network is high.

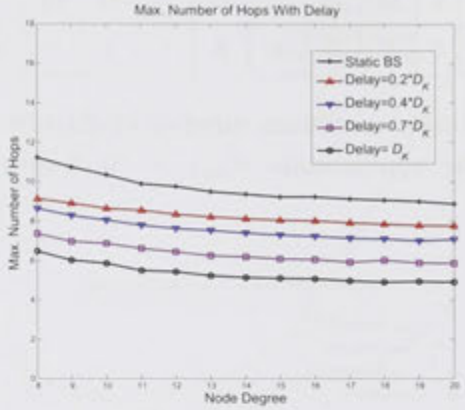


Figure 4.16: The maximum number of hops as it varies with the node degree, for static and mobile BSs.

4.6 Conclusion

In this chapter we dealt with the problem of data gathering in the mobile BS environment subject to the sensing data needed to be gathered at a specified delay. We analyse a clustering-based heuristic algorithm for finding a trajectory of the mobile BS to balance the energy consumption among sensor nodes. The algorithm allows the BS to visit all cluster heads within a specified delay. We show how to choose the number of clusters to ensure there is no packet loss as the BS moves between clusters for data gathering. We provide an analytical solution to the problem in terms of the speed of the mobile BS. We also provide analytical estimates of the unavoidable packet loss as the network size increases. Simulation is performed to evaluate the performance of the proposed algorithm against the static BS case and to evaluate the distribution of energy consumption among the cluster heads. The result has shown that, when incorporated with clustering, the use of clustering with a mobile BS can increase the network lifetime significantly. Furthermore, the proposed solution for finding cluster heads results in a uniform balance of energy depletion among cluster heads.

In the next chapter we will analyse and estimate the hop count between a source and destination nodes. The estimation of hop count is required in the study of the effect of MRs mobility on network overhead in Chapter 6.

Chapter

5

Hop Count Estimation

5.1 Introduction

In Chapter 6, we will propose a routing scheme to provide routing paths among MRs. We will study the effect of MRs mobility on network overhead when sensor nodes are used to provide the routing paths. The analysis of network overhead is required to model the hop count between the source and destination MRs. The methods most commonly used are inaccurate for sparse networks. Therefore, in this Chapter we propose a new technique for estimating routing-path hop count.

Most of the existing multihop routing protocols try to find a path to destination with a minimum number of hops in order to improve data-delivery delay and other network performance. In order to study the effect of using sensor nodes as a relay of data packets, we need to model the number of hops between the source and destination. In addition, estimation of the true distance between the source and destination based on the observation of the hop count between them has applications in localization, communication-protocol design and other areas [10][94].

The effect of hop counts on the network performance has been intensively studied in the literature using simulation. A number of analytical models are proposed to determine the required hop count for packet transmission in multi-hop wireless networks, however, none of the models are entirely satisfying. An analytical study is needed to provide a better understanding of the effect of hop count on network performance, which can help in the design of routing protocols. We propose a hop count model that estimate the number of hops for sparse networks in addition to dense networks.

The literature reports estimates of hop count using expected hop progress, where the measure of progress in Euclidean distance from source to destination per

communication hop is used to estimate the distance between a pair of nodes. One of the important factors that determines expected hop progress is the determination of next-hop zone which is the region of neighbour-node locations that could be considered as next hop for forwarding packets between source and destination.

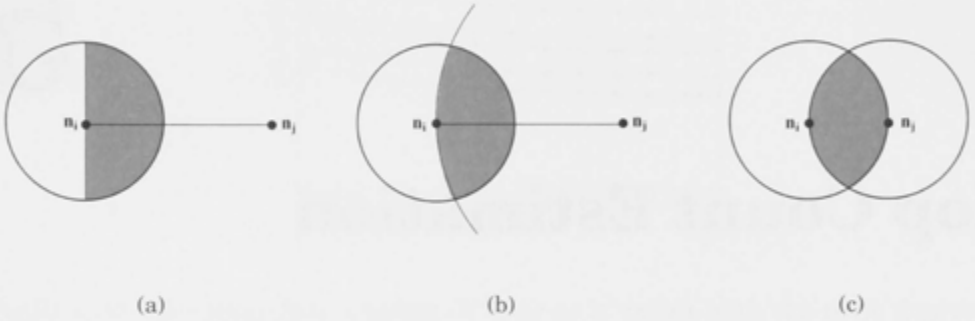


Figure 5.1: Comparison of next hop zone proposed in the literature. n_i represents the source or intermediate node and n_j represents the destination or next-hop node. (a) Hou at al. in [7]. (b) Kleinrock and Silvester in [8] and Kuo and Liao in [9] (c) Wang at el. in [10].

Hou at al. in [7] assume half of the area covered by the source transmission range in the direction of destination as the next hop zone. Kleinrock and Silvester in [8] calculate the expected progress in one hop based on assuming the next hop zone as the overlapping area bounded by the transmission range of source node and a circular area centered at destination (or intermediate node) and with radius equal to the distance between source and destination. The authors show that the proposed model does not represent true progress for small values of node degree due to network connectivity limitations. The same next-hop zone is assumed by Kuo and Liao in [9] and proposes a probability for path connectivity which represents the probability that all the hops are located in the next hop zone of the path between source and destination. Wang at el. in [10] assume the next-hop zone as the overlapping area between two circles with the transmission range as the radius and distance between the centre of these two circles. However, we show in this research that, while this assumption of next hop zone is correct for a dense network, if the nodes are sparsely deployed in the sensing field it is required to consider a bigger next-hop zone. Fig. 5.1 shows a comparison of next hop zones proposed in the literature.

Another factor that effects hop progress is the selection of the next hop node (neighbour node). Generally, there are two methods of neighbour selection for forwarding a packet to its destination, the choice of which depends on the routing protocol used. Geographic routing protocols select the next-hop node in a greedy

manner, where a node relays a packet to its neighbour that is geographically nearest to the destination. This simple relaying mechanism could cause a large number of hops compared to other routing schemes, such as proactive, reactive, and hybrid routing protocols that allow the destination node to select the shortest path. Kuo at el. in [9] and Zorzi and Rao in [95] calculate the hop count by assuming that the neighbour node that is closest to the destination is selected as the next relaying node in the path. However, the path obtained does not represent the shortest path between the source and destination when sensor nodes are sparsely deployed in the sensing field. Instead, we consider the hop progress when the shortest path between the source and destination is selected. In order to correctly capture the situation for a sparse network we examine the selection of the next neighbour node as a relaying node for the next hop.

Other research evaluates statistical relationships between hop count, h, and source-destination distance, L_{SD}, finding the conditional probability, $Pr(h|L_{SD})$, that two nodes apart at L_{SD} distance can communicate in exactly h hops. Miller in [96] and Bettstetter at el. in [97] analysed the probability of the number of hops when $h = 1$ and $h = 2$ by considering the location of nodes that are connected by a one or two hop path. Dulman at el. in [98] and Mao at el. in [99] proposed an approximation for $Pr(h|L_{SD})$ for $h > 2$. However, this literature considers dense networks only.

In this chapter, we evaluate the expected hop count between the source and destination nodes. To calculate the number of hops in the path, we first determine a distribution describing the remaining distance from next-hop (neighbour) node to destination. We then calculate the probability of selecting each of the neighbour nodes. The expected number of hops needed to cover the remaining distance is calculated by obtaining the expected progress towards the destination achieved by each hop when a path exists between them, and combined with the probability that there is a path between the source and the destination. We show that the path probability is equal to one for a dense network (high node degree), but significantly less if the node degree is small.

5.2 Network Model

We assume a WSN with n sensor nodes independently deployed with uniform distribution in a sensing field of area A. In such a network the average number of neighbours (node degree) is $d = n\pi r^2/A$, where r is the sensor node transmission range. Data packets are transmitted toward the destination using multi-hop communication and all sensor nodes are assumed to be homogeneous,

omnidirectional, and stationary. This scenario can also be regarded as describing a snapshot of mobile sensor networks.

We assume the number of network nodes is varying such that the WSN could be a sparse network with ($d = 2$, say) or a dense network. In most scenarios for WSNs, the node density is assumed to be larger of $d > 7$. However, for the sake of generality, we consider sparse in addition to dense networks. In the following section we conduct experiments to study the effect of node degree on hop count.

5.3 Exploration

In this section we conduct experiments through simulation with MATLAB to demonstrate the issues that determine hop count. We assume that the sensor nodes in the network are randomly deployed with uniform distribution in the sensing field of ($30r$ x $30r$), where r is the node transmission range. We vary the number of nodes in the network to effect change in the node degree, which is a metric of node density. The source and destination nodes are placed such that the distance between them, L_{SD}, is $4r$ or $8r$. In order to minimize the effect of the boundary of the sensing field we placed the source and destination nodes as close as possible to the center of the sensing field. The shortest path between the source and destination node is found using the Breadth First Search (BFS) algorithm where the destination node is the root of the routing tree.

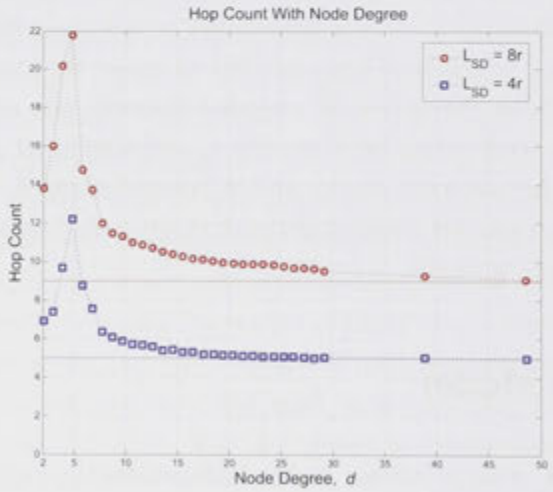


Figure 5.2: Hop count as it varies with different node degree and distance between the source and destination.

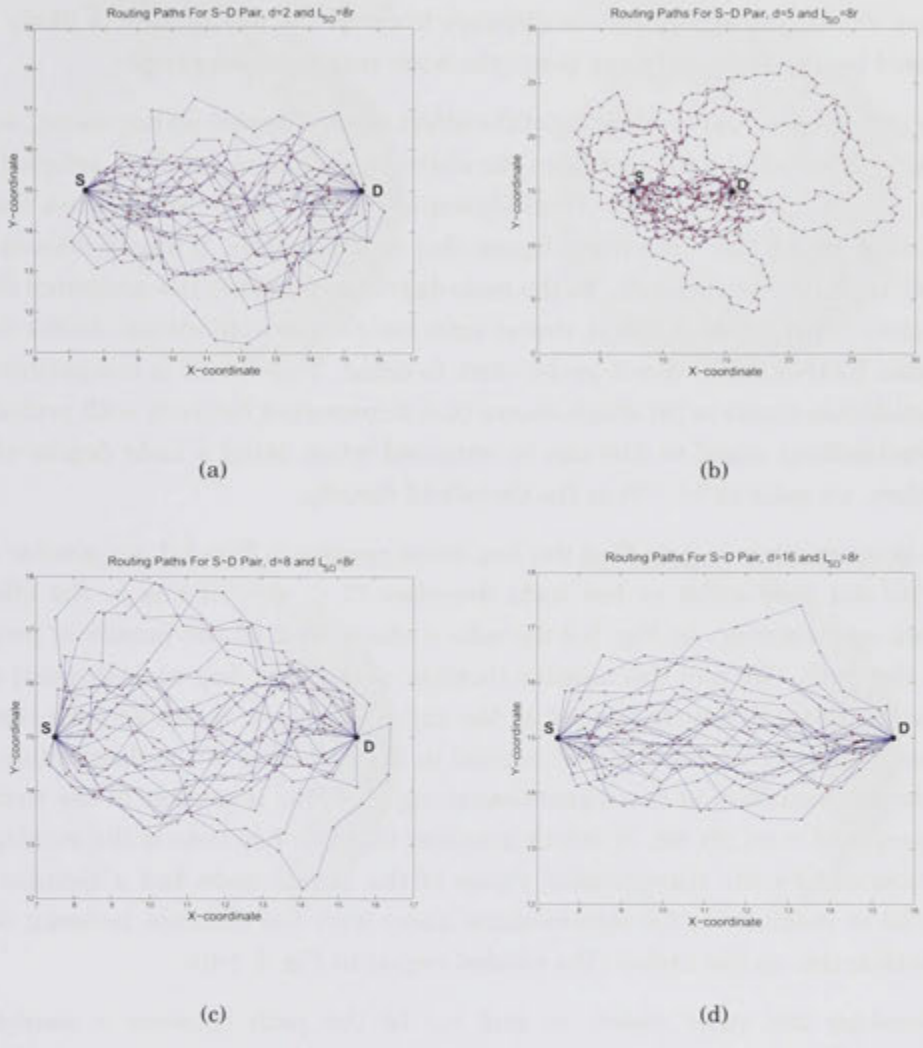


Figure 5.3: Simulation results of 25 randomly selected paths between source, S, and destination, D, nodes. Nodes are deployed with uniform distribution in the sensing field, ($30r$ x $30r$), the source and destination nodes placed at $(7.5r, 15r)$ and $(15.5r, 15r)$, respectively. $L_o = 8r$ and $r = 1$ unit length. (a) $d = 2$. (b) $d = 5$. (c) $d = 8$. (d) $d = 16$.

For each instance of deployment, network connectivity is tested to ensure there are no partitioned subnetworks. The simulation is repeated in order to obtain 100 connected networks. Fig. 5.2 shows the hop count between the source and destination nodes as it varies with node degree. It can be seen that the hop count is small for small node degree and increases with increasing node degree to reach its maximum value when the node degree is equal to five. The hop count then decreases with increasing node degree to reach its minimum value $L_{SD}/r + 1$. The

minimum hop count is achieved when all nodes in the path lie along a straight line. In other words, the accumulative distance towards the destination is likely to be increased by approximately one times the node transmission range.

To gain greater understanding of the effect of node degree on hop count, we plot 25 randomly selected paths between the source and destination for a selected node degree, $d = 2, 5, 8, 16$ at $L_{SD} = 8r$ as shown in Fig. 5.3. The results show that for small node degree there is a high chance that no path exists. If a path does exist, it is likely to be relatively direct. As the node degree approaches the minimum for full connectivity (threshold density), routes exist but they are circuitous. As the degree increases further, more direct paths start to occur. This result is compatible with the simulation result in [8] which shows that a connected network with probability of connectedness equal to 0.95 can be obtained when using a node degree of five. Therefore, we refer to ($d = 5$) as the threshold density.

It is interesting to note that the hop count results in Fig. 5.2 are similar to [9] and [10] but they differ at low node densities ($2 \leq d \leq 5$) due to the effect of network connectivity. In Fig. 5.4 we take a closer look at the details of progress along the path. We plot the relative location of the next hop at each point along the path (destination node located at the right side of n). It can be seen that, for high node density the next hop is located in the direction of the destination, and close to the boundary of the transmission region. This is similar to the next-hop zone proposed in [8, 95, 98, 9], which assumes the next-hop zone is the overlapping area bounded by the transmission range of the source node and a circular area centered at destination (or intermediate node) with the distance between source and destination as the radius (the shaded region in Fig. 5.1(b)).

Consider two relay nodes, n_i and n_j, in the path between a source and destination shown in Fig. 5.5. Assume the distance between them is slightly greater than r so that they cannot reach each other directly. Each node needs at least two neighbour nodes to keep the path connected, one is for the predecessor hop and the other is for the successive hop. There are $d - 1$ neighbour nodes that can provide a path to the next hop.

Let $Cov(p, r)$ denote a circular region centred at position p with radius r. The intersection of the transmission coverage of node n_i and n_j, is $A_1 = Area(Cov(n_i, r) \bigcap Cov(n_j, r))$ as indicated in Fig. 5.5(a). Then, the probability that there is an effective neighbour node in the intermediate area, is $P_1 = A_1/A_{i,j}$, where $A_{i,j} = Area(Cov(n_i, r) \bigcup Cov(n_j, r))$, and the probability that there is no one-hop (direct) path is $(1 - P_1)^{d-1}$.

If n_i cannot be reached within a single intermediate hop, there may still be a path connecting n_i and n_j. We assume that such a path exists if both n_i and n_j have

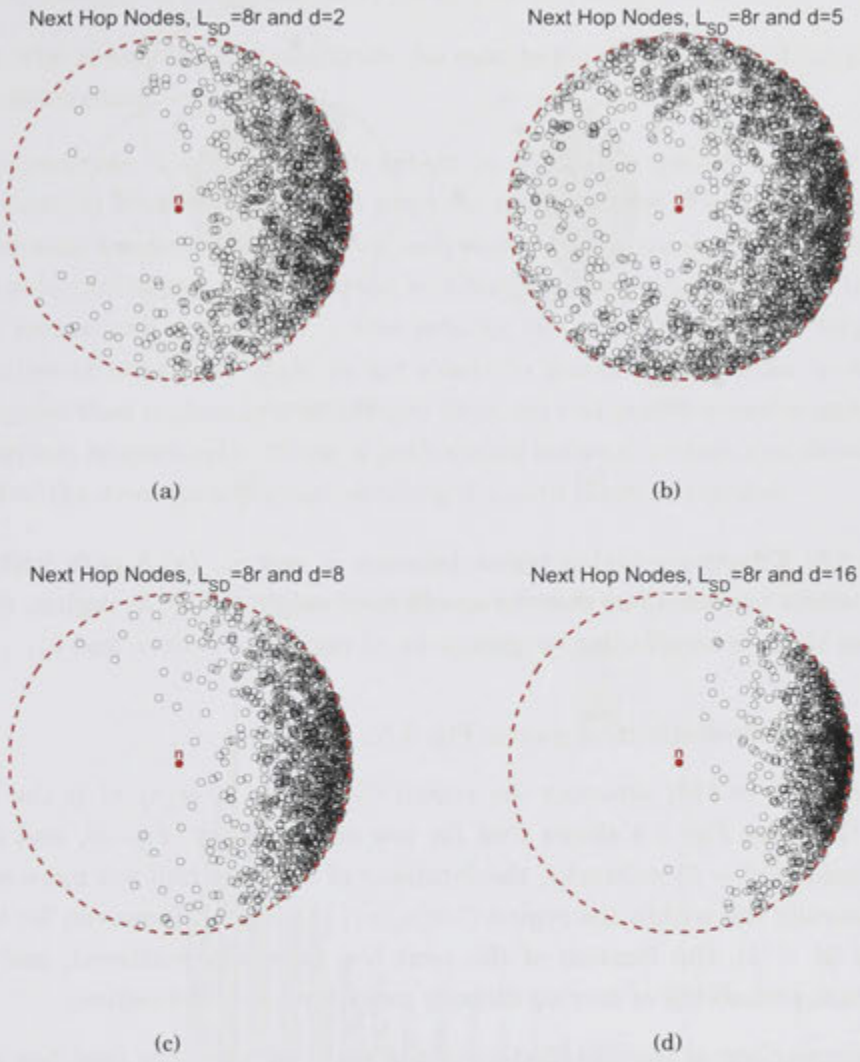
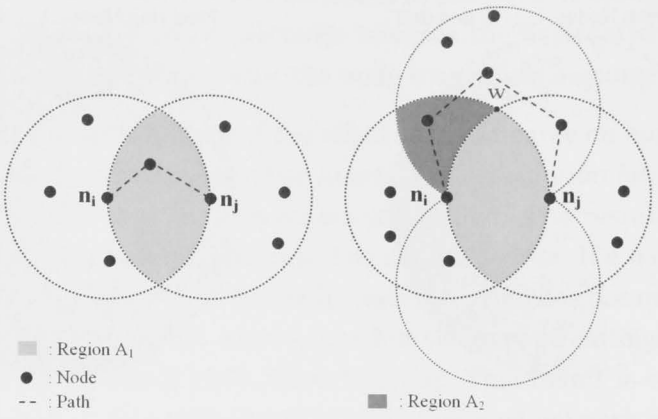


Figure 5.4: Simulation results of the next hop of 100 paths between source, S, and destination, D, nodes. Nodes are deployed with uniform distribution in the sensing field, ($30r$ x $30r$), the source and destination nodes are placed at $(7.5r, 15r)$ and $(15.5r, 15r)$, respectively. $L_o = 8r$ and $r = 1$ unit length. (a) $d = 2$. (b) $d = 5$. (c) $d = 8$. (d) $d = 16$.

neighbours in the regions $A_2(n_i, w) = Area(Cov(n_i, r) \bigcap Cov(w, r))$ and $A_2(n_j, w) = Area(Cov(n_j, r) \bigcap Cov(w, r))$ respectively, where w is indicated in Fig. 5.5(b). If there are no neighbours in these areas, the chance there is a path with two-hops between n_i and n_j is very low.

Then, $P_2 = (2A_2/\pi r^2)^2$ is the probability that there is a neighbour node in the regions $Cov(n_i, r) \bigcap Cov(w, r)$ and $Cov(n_j, r) \bigcap Cov(w, r)$ (or the equivalent areas



(a) (b)

Figure 5.5: Effective relaying region between n_i and n_j. (a) A path with a single intermediate hop exists as there is an effective neighbour in A_1 region. (b) A path exists as there is an effective neighbour in A_2 region for both n_i and n_j.

below the source-destination pair in Fig. 5.5).

The work in [10] assumes the region $Cov(n_i, r) \bigcap Cov(n_j, r)$ is the next-hop zone. However, Fig. 5.4 shows that for low node density ($d = 2$), and moderate node density ($d = 8$) networks, the locations of the next hop are more scattered, but generally fall within the region $Cov(n_j, r) \bigcap Cov(w, r)$. However, for threshold density ($d = 5$), the location of the next hop is widely scattered, and there is significant probability of moving directly away from the destination.

A comparison of the distribution of the angle between the next-hop node and destination at different node densities is shown in Fig. 5.6. The result shows that next hop in exactly the opposite direction to the destination occurs at the threshold density only. The result also shows that as node density increases, the probability of selecting the next hop in the direction of destination increases. The selection of the next-hop node has a significant effect on hop progress as shown in Fig. 5.7 where hop progress is expressed in proportion to the transmission range. The results show that a high probability of negative progress occurs at threshold density and the probability of achieving positive progress increases with node density.

In this chapter, we introduce a hop progress model with arbitrary node degree and distance between source and destination. This model is necessary to calculate the hop count. Before we start the hop count analysis, we need to describe and define the following parameters (refer to Fig. 5.8)

- Let n_i be the i-th neighbour node for any sensor node ($i = 2, ..., d$).
- $L_{r,i}$: The remaining distance from the next hop node (i-th neighbour node) to the destination.
- Hop progress, $L_{SD} - L_{r,i}$: The extent to which the packet is closer to the destination after the hop. Hop progress ranges from $-r$ to r where r is the maximum transmission range of each node. Hop progress is determined by the selection of the next hop node and it effects the number of hops between the source and destination. The smaller the hop progress the larger the number of hops that each packet needs to traverse. We refer to the hop progress that is greater and smaller than zero as positive and negative hop progress, respectively. When a path exists between source and destination nodes, the average hop progress along the path must be positive.

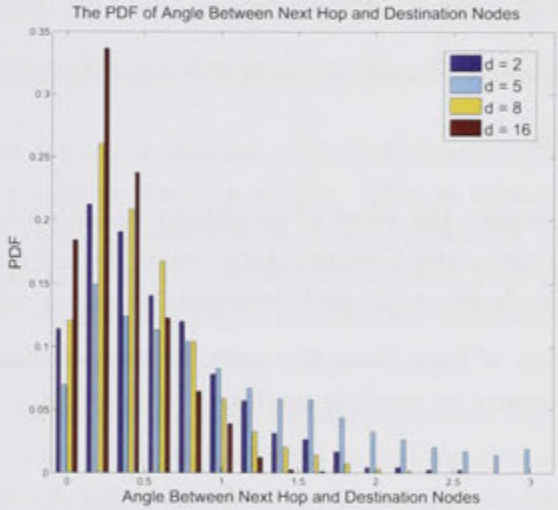


Figure 5.6: The PDF of the angle between the next hop and destination nodes at different node degrees, $L_{SD} = 8r$.

5.4 Expected Hop Progress

In order to calculate the expected hop count, we need to consider both hop progress and connectivity effects. This approach differs from common approaches which consider only hop progress and do not correctly address how the hop count is effected at low-node density.

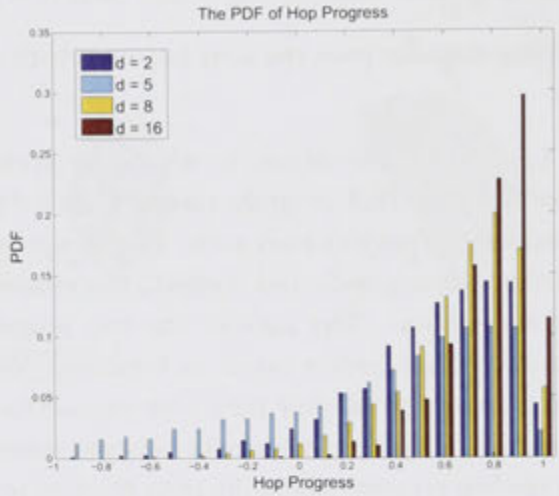


Figure 5.7: The PDF of normalized hop progress ($r = 1$) at different node degrees, $L_{SD} = 8r$.

5.4.1 Hop Length

In this section we consider the effect of neighbour nodes on hop progress in one hop. In Fig. 5.8, n_s is the source having a data packet destined to n_d (in fact n_s can be any intermediate node along the path between the source and destination).

To find the number of hops along the path, we assume that each node in the network has d neighbours at random angles, $\theta_i \in [0, 2\pi]$, $i = 1, ...d$ and random distances, $d_{o,i} \in [0, r]$.

Assuming that the n nodes in the network are uniformly distributed in a field of area A, then the average number of network neighbours is $d = n\frac{\pi r^2}{A}$. Let the distance from n_s to n_d be L_{SD}. Then the remaining distance to the destination from neighbour n_i is

$$L_{r,i} = \sqrt{L_{SD}^2 + d_{o,i}^2 - 2L_{SD}d_{o,i}\cos(\theta_d - \theta_i)} \tag{5.1}$$

The PDF and CDF of the remaining distance to the destination are $f_{L_r}(l_r)$ and $F_{L_r}(l_r)$ and are given in Appendix A. The hop progress is determined by the selection of the next hop node (i-th neighbour node). In the following we analyse the effect of neighbour-node selection as the next hop node on the distance between the neighbour node and destination in order to determine the hop progress.

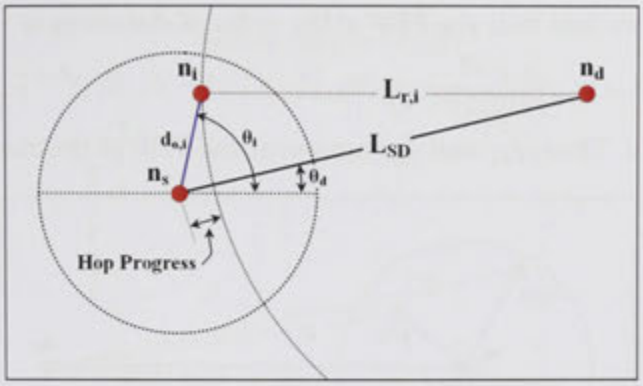


Figure 5.8: An example illustrates the distance relationship between source node, n_s, neighbour node, n_i and destination node, n_d. n_i is neighbour to n_s at distance $d_{o,i}$ and angle θ_i. L_{SD} is the distance between n_s and n_d, θ_d is the position angle of n_d. $L_{r,i}$ is the remaining distance to destination after selecting n_i as the next hop. $L_{SD} - L_{r,i}$ is the hop progress.

5.4.2 Effect of Neighbour Choice on Hop Progress

In this research, we assume a shortest path with minimum number of hops exists between the source and destination nodes. This is achieved by using flooding schemes proactive, reactive, and hybrid routing protocols. As mentioned earlier this assumption differs from Kuo at el. in [9] and Zorzi and Rao in [95] who calculate the hop count assuming that the nearest neighbour node to destination is selected as the next hop node.

In order to calculate the distance between the next hop and the destination nodes for different values of d, we first need to find the distribution of the distance by selecting neighbour nodes at different distances to the destination. Let $L_{r,1}$, $L_{r,2}$,..., $L_{r,d}$ denote a set of d random distances to the destination drawn from the CDF, $F_{L_r}(l_r)$ given in Appendix A. Let L_r^1, L_r^2,..., L_r^d denote a rearrangement of these distances such that $L_r^1 < L_r^2 <, ..., < L_r^d$. For instance, $L_r^1 = \min\{L_{r,1}, L_{r,2}, ..., L_{r,d}\}$ is the smallest of the distances and is corresponding to select the closest neighbour node to the destination when selected, and $L_r^d = \max\{L_{r,1}, L_{r,2}, ..., L_{r,d}\}$ corresponds to the farthest neighbour node from the destination. Moreover, let the nodes n_1,... ,n_d be renamed n^1,...,n^d according to their distance from the destination, so that n^1 and n^d are the closest and farthest neighbour nodes to the destination, respectively. Fig. 5.9 shows an example of four neighbour nodes ordered according to distance from the destination. Along the path, the positions of neighbour node are independent and identically distributed in angle and distance from n_s. Using

order statistics, we find that the PDF of the order of distances is

$$f_{L_r^j}(L_r) = \frac{d!}{(j-1)!(d-j)!}(F_{L_r}(l_r))^{j-1}(1-F_{L_r}(l_r))^{d-j} f_{L_r}(l_r), \tag{5.2}$$

where $j = 1, ..., d$. Thus, $f_{L_r^1}$ and $f_{L_r^d}$ represent the PDF of the closest and farthest

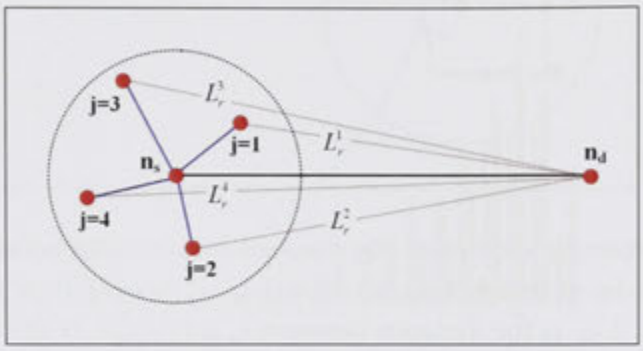


Figure 5.9: An example illustrates the order of neighbour nodes according to its distance to destination for $d = 4$.

distance to the destination, respectively. The expected value of the remaining distance between the neighbour nodes and the destination for the j^{th} order distance is

$$E\{L_r^j\} = \int_{L_{SD}-r}^{L_{SD}+r} L_r f_{L_r^j}(L_r) dL_r, \tag{5.3}$$

then, the expected hop progress towards the destination by each hop is

$$E\{\text{Hop progress}|\text{order } j \text{ distance}\} = L_{SD} - E\{L_r^j\}. \tag{5.4}$$

Using (5.4), we can study the effect of the order of neighbour nodes on the expected hop progress as shown in Fig. 5.10 where the normalized hop progress is plotted for a different number of neighbours.

It can be seen that the expected hop progress increases with the number of neighbours when the closest node to destination is selected ($j = 1$). In addition, negative progress is obtained (the path goes further away from the destination) when selecting neighbours at high order for a particular d.

Selecting the closest neighbour node to destination increases the hop progress and thus decreases the number of hops required to reach the destination as assumed in [9] and [95]. However, the path obtained in this way may not represent the shortest path with a minimum number of hops between the source and destination at low node density, since the nearest node to the destination may lead to other hops that goes away from the destination. Instead, we assume that hop progress is a result of selecting neighbour nodes at different orders. Each order of neighbour node is selected based on calculated probability as illustrated in the following section.

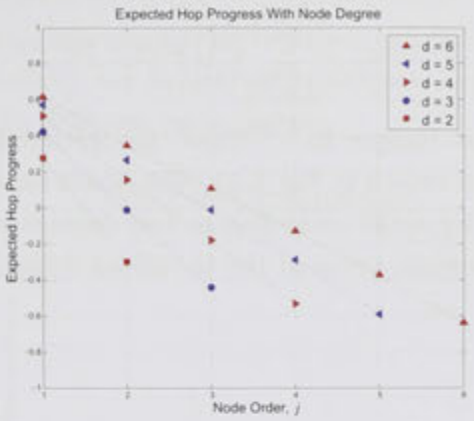


Figure 5.10: Expected hop progress, from (5.4) as it varies with the order of neighbour (distance to destination), for different node degree when $r = 1$.

5.4.3 Neighbour Selection Probability

In accordance with the simulation results in Section 5.3, we assume that n^1 is selected as the next hop only when the node degree is greater than six while at low node densities ($2 \leq d \leq 6$) the selection of n^1 is based on a certain calculated probability determined below.

In order to determine the probability of selecting n^1 as the next hop, we need first to define next-hop zone, which is the location of neighbour nodes that are possible to be selected as the next hop for forwarding packets between the source and destination. Referring to Fig. 5.4 that shows the position of next hop nodes relative to destination, we assume that the next-hop zone is equal to $A_1 + 2A_2$ as indicated in Fig. 5.5 as this zone covers most of the cases of node degree. Referring to Fig. 5.5, the ratio of $A_1 + 2A_2$ to $r^2\pi$ is equal to $P = 0.724$. The probability that a node will present in the next-hop zone is equal to P. Thus for d node degree, the number of sensor nodes in the next-hop zone follows the binomial distribution $n_z \sim B(d, P)$.

When multiple neighbours lie in the next-hop zone, it is likely to select n^1 since most of the neighbour nodes can access the same next hop nodes. Therefore we assume that the probability of selecting n^1 is equal to the probability that the number of nodes in the next-hop zone is equal to or greater than the expected number of nodes ($E\{n_z\} = dP$) in the next hop zone. Thus, the probability of selecting n^1 is given by

$$Pr(n^1|d) = \sum_{i=\lfloor dP \rfloor}^{d} \binom{d}{i} P^i (1-P)^{d-i}, \tag{5.5}$$

where $[x]$ is the nearest integer to x. Then, the probability of selecting n^1 for different node degrees is shown in Fig. 5.11. The result shows that the probability of selecting n^1 is high for small node degree and decreases with increasing node degree to reach its minimum value at the threshold density, then the probability increases at $d = 6$.

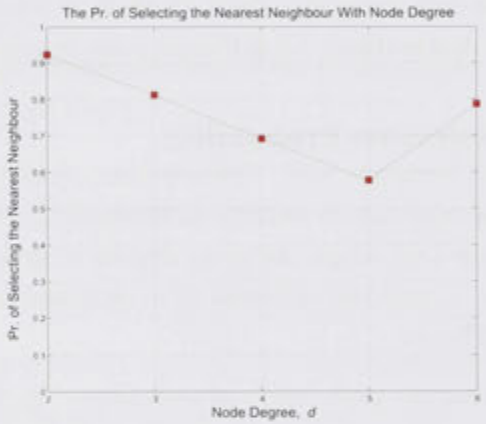


Figure 5.11: The probability of selecting the nearest neighbour node to destination, from (5.5) as it varies with node degree at $2 \leq d \leq 6$.

In order to determine the probability of selecting other neighbour nodes as the next hop, the geometric distribution is used with the probability of success equal to $Pr(n^1|d)$. The geometric distribution is used to find the probability of the first success occurs after j trails. Because the neighbours are independent and identically distributed, the probability of selecting the second nearest neighbour node to destination is equal to the probability of selecting the nearest neighbour node to destination occurs next and so on. Thus, the neighbour selection probability is given by

$$Pr(n^j|d) = (1 - Pr(n^1|d))^{j-1} Pr(n^1|d), \tag{5.6}$$

where $j = 1, ..., 6$ represents the order of neighbour node selection. For instance, $j = 1$ and $j = d$ represent the nearest and farthest neighbour node to destination, respectively. Fig. 5.12 shows the neighbour selection probability as it varies with neighbour node order for different node degrees.

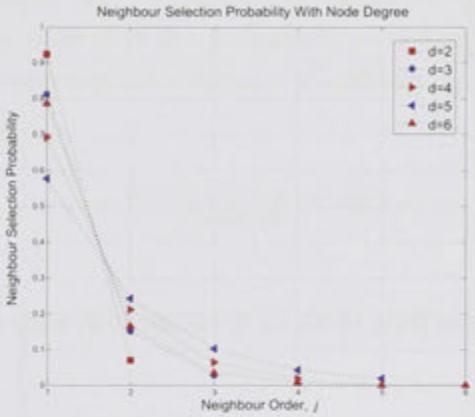


Figure 5.12: The probability of selecting the neighbour node, from (5.6) as it varies with the order of neighbour node for different node degree at $2 \leq d \leq 6$.

Using (5.4) and (5.6), and assuming the network is fully connected, the expected hop progress can be expressed by

$$\begin{aligned} & E\{\text{hop progress}|\text{connected network}\}(d) \\ &= \sum_{j=1}^{d} E\{\text{Hop progress}|\text{order } j \text{ distance}\} Pr(n^j|d) \\ &= \sum_{j=1}^{d} (L_{SD} - E\{L_r^j\}) Pr(n^j|d), \qquad 2 \leq d \leq 6 \end{aligned} \tag{5.7}$$

In summary, the expected hop progress for different node degree is given by,

$$\begin{aligned} & E\{\text{hop progress}|\text{connected network}\}(d) \\ &= \begin{cases} \sum_{j=1}^{d} (L_{SD} - E\{L_r^j\}) Pr(n^j|d) & 2 \leq d \leq 6 \\ L_{SD} - E\{L_r^1\} & d > 6. \end{cases} \end{aligned} \tag{5.8}$$

The effect of the number of neighbours and the distance between the source and destination on the hop progress is shown in Fig. 5.13 where hop progress is plotted for different node degrees. It can be seen that the hop progress generally increases with node degree except when the number of neighbours is equal to 4 and 5. It can be noticed that the maximum increases in hop progress occurs when $d = 6$ and $d = 7$.

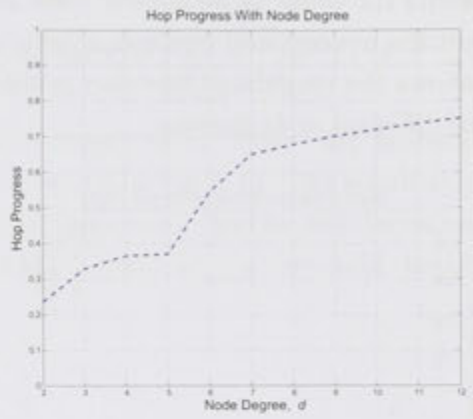


Figure 5.13: Hop progress from (5.8), as it varies with node degree when $L_{SD} = 8r$ and $r = 1$.

5.4.4 Probability of Network Connectivity

The derivations so far have been based on the assumption that a path exists from source to destination (i.e., a high a node density network). However if the average number of network neighbours is small, the probability of a path existing between a particular source-destination pair is small. We consider the random location of neighbours in order to derive the probability of connectivity from the source to destination for different node degrees.

Refering to Fig. 5.5, assume the distance between n_i and n_j is slightly greater than r so that they cannot reach each other directly. Intuitively, there is a path if there is at least one neighbour node in the overlap area of their transmission coverage, or a path around the overlapping such as illustrated by Fig. 5.5 (b).

The probability that there is an effective neighbour node in the intermediate area, is $P_1 = A_1/A_{i,j}$, where $A_{i,j} = Area(Cov(n_i, r) \bigcup Cov(n_j, r))$, and the probability that there is no one-hop (direct) path is $(1 - P_1)^{d-1}$. $P_2 = (2A_2/\pi r^2)^2$ is the probability that there is a neighbour node in the regions $Cov(n_i, r) \bigcap Cov(w, r)$ and $Cov(n_j, r) \bigcap Cov(w, r)$ (or the equivalent areas below the source-destination pair in Fig. 5.5).

The probability that the next relay cannot be reached using more than one hop is $(1 - P_2)^{(path)}$, where $(path)$ is the number of possible paths between n_i and n_j. We assume the number of intermediate relays is equal to two and thus the maximum number of possible paths is equal to $(d - 1)^2$.

We propose that if the network connectivity is low there is no path with one or

two intermediate relays between n_i and n_j. Then the connectivity probability, P_c, is given by

$$P_c(d) = 1 - ((1 - P_1)^{d-1}(1 - P_2)^{(d-1)^2}). \tag{5.9}$$

Using (5.9), we can study the effect of node degree on network connectivity as shown in Fig. 5.14, where the probability of connectivity is plotted for different node degrees. It can be seen that the probability of connectivity is small for small node degrees and increases with the increase of node degree, to reach the maximum value when $d = 6$.

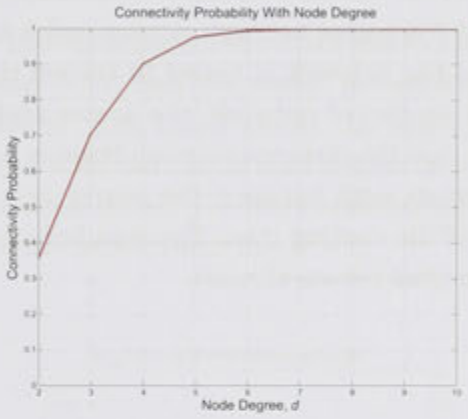


Figure 5.14: Connectivity probability as it varies with different node degrees, from (5.9).

5.4.5 Expected Hop Count

In order to calculate the average number of hops required to reach the destination, we need to estimate the hop progress considering the effect of network connectivity. Using (5.8) and (5.9) the expected hop progress can be expressed by

$$\begin{aligned} & E\{\text{hop progress}\}(d) \\ & = E\{\text{hop progress}|\text{connected network}\}(d)Pr\{\text{connectivity}\} \\ & = \begin{cases} \sum_{j=1}^{d}(L_{SD} - E\{L_r^j\})Pr(n^j|d)P_c(d) & 2 \le d \le 6 \\ (L_{SD} - E\{L_r^1\})P_c(d) & d > 6. \end{cases} \end{aligned} \tag{5.10}$$

Then, the hop count is given by

$$E\{hop = h\}(d) = \frac{L_{SD}}{E\{\text{hop progress}\}(d)}. \tag{5.11}$$

5.5 Results

In this section we present and discuss results of theoretical calculations and Monte-Carlo simulations, conducted in MATLAB, for an expected hop count given in (5.11).

In the simulation, nodes are uniformly distributed in a circular sensing field in order to minimize the boundary effect. The radius of the sensing field is $R = 30r$ and the maximum transmission range (r) of each node is set to $250\ m$. We varied the number of sensors in the network in order to emulate the change in the node degree. To obtain a fully connected network, sensor nodes are repeatedly deployed in the sensing field and the network is tested to ensure there are no partitioned subnetworks. For each connected network, the source and destination nodes are randomly selected such that the distance between them is L_{SD}. The BFS algorithm is used to find the shortest path between the source and destination where the destination is the root of the routing tree. The simulation results shown here are averaged over 100 (connected network) trials.

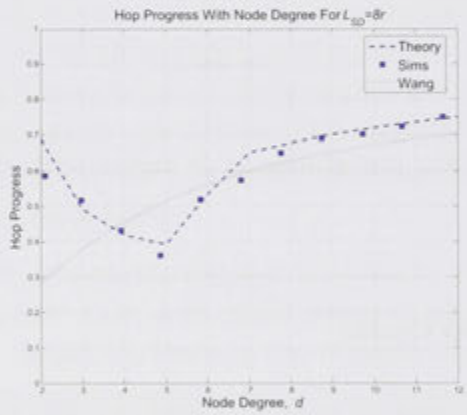


Figure 5.15: Hop progress as it varies with node degree, theoretical results from (5.8) and Wang et al. in [10]. Simulation results are depicted by markers while theoretical results are depicted by lines.

Fig. 5.15 compares the theoretical and simulation hop progress. The theoretical hop progress is calculated using (5.8) and it is also compared with the hop progress model proposed in [10]. The result shows that hop progress is large for a small node degree and decreases with increasing node degrees to reach its minimum value when the node degree is equal to five. The hop progress then increases with increasing node degree. The reason for the high progress for small node degree is network connectivity issues. For a small node degree there is a high chance that

no path exists, however, whenever a fully connected network is obtained, the path between source and destination is likely to be relatively direct. As the node degree approaches the minimum for full connectivity, routes exist but they are circuitous and as the node degree increases further, more direct paths start to occur.

The result also shows that our model for calculating the expected hop progress matches the simulation results while the curve of the expected hop progress proposed by Wang et al. in [10] only approaches the simulation hop progress for high node degree. Moreover, it behaves differently at low node degree due to the effect of network connectivity.

Fig. 5.16 compares the theoretical and simulation hop count. The expected hop count is calculated using (5.11) and the model proposed by Wang et al. in [10] is included for comparison. The result shows the hop count is small for small node degree and increases with increasing node degree to reach its maximum value when the node degree is equal to five. The hop count then decreases with increasing node degree to reach its minimum value $L_{SD}/r + 1$.

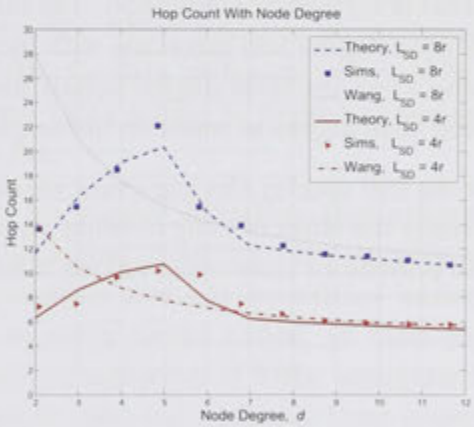


Figure 5.16: The hop count as it varies with node degree, theoretical results from (5.11) and Wang et al. in [10]. Simulation results are depicted by markers while theoretical results are depicted by lines.

The result shows that our model matches well with the simulation hop count

for different node degrees, while the model proposed by Wang et al. tries to match the simulation for high node degree $d \geq 6$.

5.6 Conclusion

In this chapter we proposed an analytical model to estimate the hop count between source-destination pairs in a wireless network with an arbitrary node degree when the network nodes are uniformly distributed in the sensing field. The relation between the Euclidean distance between the source-destination pairs and the hop count is useful in many ad hoc and sensor network applications, such as localization, communication-protocol design, estimating end-to-end delay and power consumption along the path.

We consider both the hop progress and connectivity effect on hop count. Our approach differs from common approaches which consider only hop progress and do not correctly address how the hop count effects at low node densities.

The shortest path with a minimum number of hops is assumed. This can not be achieved by selection of the nearest neighbour nodes to destinations at low node density, so we calculate the probability of selecting neighbour nodes as the next hop node. The analytical model is verified by simulation. The results show that the hop count is small for small node degree and increases with increasing node degree to reach its maximum value when the node degree equals five. The hop count then decreases with increasing node degree to reach its minimum value.

In the next chapter we will employ the hop count results from this chapter in the analysis of MRs to study the effect of their mobility on network overhead when sensor nodes are used to provide an interaction among MRs.

Chapter

6

Routing Scheme For Mobile Relays

6.1 Introduction

In Chapter 4, we show that the use of a mobile instead of static BS for data gathering can improve network lifetime. However, we demonstrate a single mobile BS is insufficient to achieve the required network lifetime and avoid packet losses in the case of large networks. There has been an increased research focus on the use of mobile relays (MRs) for data gathering, where MRs roam the sensing field and collect sensing data using single or multi-hop communication and forward aggregated data to the BS.

In Section 2.2.2.2, we survey literature that reports the use of MRs for data gathering. We show that MRs can interact with each other using short or long range communication and are sometimes assumed to have the same communication range as static sensor nodes. In most cases the literature does not consider the energy consumption of MRs, assuming the MRs are equipped with powerful batteries or they can replenish their energy periodically. On the other hand, the energy cost of MRs is considered in other research, as the energy replenishment of MRs cannot always be possible due to the constraints of the physical environment. One approach is to restrict the mobility of MRs so that MRs maintain a connected path with each other and with the BS all the time while they are moving, and sensing data is forwarded to the BS through the MRs [88].

In this Chapter we assume a number of MRs roam through the sensing field and sensor nodes are used to provide interaction among MRs in order to reduce the energy consumption of MRs and to allow them to move freely in the sensing field. The use of MRs for sending data via static sensor nodes forms a hybrid of mobile

ad hoc and sensor networks, which we refer to as mobile ad hoc sensor network (MAHSN). In a MAHSN sensor nodes are static and send data packets to the MRs only, whereas in mobile ad hoc networks all nodes move arbitrarily and any two nodes can be the source and destination of data packets. In addition an MAHSN differs from a WSN in that sensor nodes have to be involved in providing routing paths among MRs. The mobility of MRs inevitably causes additional overhead in routing protocols. The overhead may offset the benefit brought by mobility.

Most of the sensor networks research involving MRs proposes routing protocols that deal with finding the routing path between the MRs and sensor nodes for data gathering to balance the energy consumption among the sensor nodes [100, 101, 102, 103]. However, in this work we are interested in finding a routing scheme that provides an interaction among the MRs via sensor nodes that minimizes network overhead. To the best of our knowledge, little attention has been paid on the scenario that the MRs use sensor nodes for sending data packets among them and to the BS.

In this Chapter, we propose a routing scheme based on the multipath DSR protocol. We explore how the number of paths and spread of neighbour nodes used by the source MR to reach the destination affects the network overhead. An analytical model of network overhead is developed and verified by simulation. In this research, we refer to the source and destination MRs as n_S and n_D, respectively.

In the following section we introduce the principle operations of the multipath DSR routing protocol.

6.2 Multipath DSR Routing Protocol

In the conventional DSR routing protocol, a source node attempts to discover a route to a destination only when it is presented with a packet for forwarding to that destination. Usually a route cache is employed to avoid carrying out a route discovery for every new packet to the same destination. If no cached route is available, the source node, n_S, initiates a new route discovery, by broadcasting a route request (RREQ) packet throughout the network. Any node receiving a duplicate RREQ discards the duplicate, so that each node is considered to have only dealt with each RREQ once. When the destination node, n_D, receives a RREQ, it returns to n_S a route reply (RREP) packet back along the route via which the RREQ arrived. In addition, if a node involved in forwarding a data packet along an established route determines that the link of which it is at the head is no longer valid, it returns an error packet (RERR) to n_S back along the route. Monitoring the

correct operation of a route in use is referred to as route maintenance [104][105].

We consider a multipath modification of DSR, where n_D stores all the copies of routes which arrive, in order to determine which requests to respond to. The source keeps all routes received via the reply packets in its route cache. Most multipath routing protocols assume disjoint paths, however the necessity of this assumption is questioned [106], particularly in the MAHSN case where intermediate nodes on the route are static. Instead, we assume only that each route starts with a different neighbour node, as then a link failure with a neighbour in one route does not effect the others. The source keeps all routes received via the reply packets in its route cache. Different criteria are suggested to determine the number of alternative routes used by the protocol [107]. We later show that the best results are achievable with about six routes.

We consider two mechanisms for the source node to obtain multiple routes to destination, passive and active multipath. First, we assume the n_S passively finds routes to n_D by for example enabling the promiscuous receive mode in which the source node configures its wireless network interface to listen to all traffic without filtering based on destination address [104]. In addition, routes can also discover when n_S forwards a route reply of other route discovery process [108].

In this mechanism, n_S opportunistically caches multiple routes to n_D. Thus there is no guarantee n_S can learn multiple routes to destination as the number of routes depends on network traffic. Therefore, we consider n_S actively discovers routes to n_D through multiple route replies generated by n_D. In this case, we calculate the expected overhead when the routes to n_D discovered through randomly selected neighbours and appropriate selected neighbours.

6.3 Network Model

We assume sensor nodes are randomly deployed in the sensing field, each node is equipped with an omnidirectional antenna and signal attenuation is due only to path loss related to distance transmitted. MRs roam the sensing field to gather sensing data using single or multi-hop communication. Each MR aggregates the received data in order to reduce the volume of data required to be sent to other MRs or to the BS. We assume the MRs have limited energy resources and it is difficult to replenish their energy periodically. Therefore, MRs use the same communication range as sensor nodes, r. We assume as in [92], MRs interact with each other through sensor nodes using multi-hop communication. All node-to-node communications are assumed to be bi-directional.

All the sensor nodes are stationary or they move at very low velocity compared to the MRs and so can be considered stationary. The MRs move with a constant velocity, v, in a direction, θ_m.

MRs and the BS are assumed to know their own locations. They can also estimate the locations of their neighbour nodes but no information can be obtained about the locations of other sensor nodes. A survey of location techniques can be found in [109].

6.4 Proposed Multipath Routing Scheme

In this section, we propose a multipath modification of DSR. Our routing scheme aims to avoid path failure between the source and destination due to links being broken between the MRs and their neighbours. In order to achieve this, we propose to augment the routes obtained by route discovery, to be through selected neighbour nodes of the source and destination MRs. The selection of neighbour nodes is based on their location relative to the MR. Thus, each MR needs to establish a neighbour table in order to select neighbour nodes for route modification.

6.4.1 Neighbour Table

Every MR advertises its presence by periodically broadcasting a special packet called a *beacon*. A sensor node, upon receiving the beacon, send its data to the MR. The MR buffers sensing data and establishes a neighbour table that stores sensor location and its network address. The information in the table is used by the MR to create a routing path between the MR and its neighbour nodes. This table must be updated frequently in order to delete the nodes that are not neighbours of the MR due to the movement of the MR.

6.4.2 Neighbour Selection

We assume different methods for neighbour selection for the source and destination MRs. The source MR initiates a route discovery when the route cache is empty. The source receives routes as determined by the destination. The first hop of each of these routes will be a neighbour. We propose that the source augments the routes to be through the neighbour nodes that is closest to the direction of movement of the source, as shown in Fig. 6.1. The selected neighbour could be several hops from the first node in the broken route. This method adds extra hops to the modified

route, however, we will see later that these extra hops reduce the overhead by avoiding frequent route discoveries.

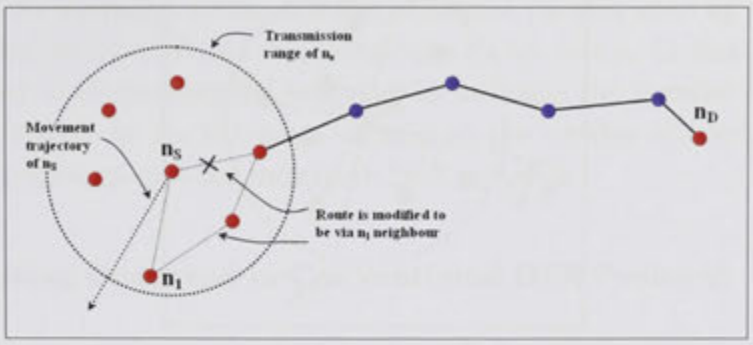


Figure 6.1: An example illustrates how the source augments the routes to be through the neighbour node that is closest to the direction of movement of the source.

A different process occurs at the destination node. The destination MR receives different copies of route requests. We propose that the destination selects multiple routes to the source through different neighbour nodes as illustrated in Fig. 6.2. Using the neighbour table, the destination MR creates routes to the source through appropriately distributed neighbours. That is, the destination MR creates routes through neighbours that are approximately separated by an angle θ (neighbours θ-separated). This can be achieved in a dense network using, for example, topology control as illustrated in Chapter 2. Most literature in the field of topology control uses neighbours θ-separated in order to reduce signal interference, however, in this research we are interested in finding routes between source and destination MRs via neighbours at different directions. This 'directional diversity' ensures a route failure due to the movement of destination MR in one direction does not effect other routes.

The destination MR aims to find neighbour nodes with separation angle $\theta = 2\pi/\sigma$, where σ is the number of paths. To achieve that, the destination MR sorts neighbour nodes in increasing order according to their angular arrangement around the destination MR. Then, for each neighbour node, it creates an axis centred at the destination MR with θ as separation angle. The set of neighbour nodes that is closest to the axis is selected and the total angle difference (error) from the axis is calculated. The set of neighbour nodes with a minimum total error is selected. Fig. 6.2 illustrates the concept of neighbour selection and Algorithm 2 describes the algorithm for neighbour selection at the destination MR.

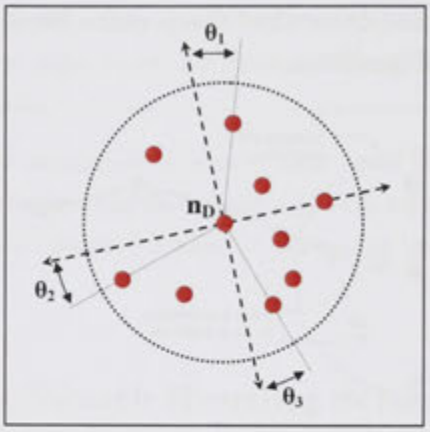


Figure 6.2: An example illustrates the selection of $\sigma = 4$ neighbour nodes at the destination MR from a total of $d = 9$ physical neighbours. The total variation of the chosen neighbours from the ideal of θ-separation is $\theta_{error} = \theta_1 + \theta_2 + \theta_3$.

Algorithm 2 Algorithm to select σ θ-separated neighbours from a set of d neighbours.

Input: σ, d

Output: Set of selected neighbours u_θ

$\theta = 2\pi/\sigma$;

$u(i)$=Set of neighbour nodes sorted in an increasing order according to their angles relative to the destination MR, $1 \leq i \leq d$;

for *each neighbour* i, $1 \leq i \leq d$ **do**

  $\theta_{error}(i) = 0$;

  for *each axis* j, $1 \leq j \leq \sigma - 1$ **do**

    θ_u=angle of $u(i)$

    $\theta_{axis} = \theta_u + j * \theta$;

    if $\theta_{axis} > 2\pi$ **then**

      $\theta_{axis} = \theta_{axis} - 2\pi$;

    end

    $uc(j)$=Closest neighbour to θ_{axis};

    θ_c=Angle of $uc(j)$ relative to the destination MR;

    $\theta_{error}(i) = \theta_{error}(i) + \mid \theta_{axis} - \theta_c \mid$;

    Add $uc(j)$ as a selected neighbour into $SN(i, j)$;

  end

end

u_θ=Set of neighbours from SN that at the i-th of the minimum $\theta_{error}(i)$;

6.5 Overhead

We refer to the overhead as the number of control packets used by the network nodes in order for n_S to obtain single/multiple routes to n_D. In this research we are interested in minimizing the overhead by selecting the number of multipath routes to be stored. In the following, we analyse the number of overhead packets generated by conventional and multipath DSR protocols.

6.5.1 Routing Overhead in Conventional DSR Protocol

DSR routing protocol assumes that the network is dynamic and routes are unlikely to remain valid in the long term. Therefore, when the details of an established route are saved in the cache, a route expiry time, T, is determined and saved for that route. After this time it is assumed that the route is no longer valid. If n_S wishes to transmit to n_D after this time it must carry out a new route discovery process. That is, if we let t_a be the time of arrival of a new packet destined for n_D, then, if $t_a < T$ and the cached route is not broken, no routing overhead is incurred in sending the new data packet. However, if $t_a > T$ a route discovery process is automatically undertaken, using flooding of RREQs. If there are n nodes in the network, $n-1$ RREQ packets are transmitted during a flood (all nodes in the network receive the RREQ except for n_S), plus h RREPs where h is the number of hops between n_D and n_S in the route chosen by n_D as decided by the route of the first RREQ to reach n_D. This scenario assumes that only one route is cached in any route discovery process.

In summary, the number of overhead packets generated by an on-demand routing protocol in response to a packet arrival at time t_a is given by,

$$\mathrm{OH} = \begin{cases} 0 & t_a < T \\ n + h - 1 & t_a > T. \end{cases} \tag{6.1}$$

Usually a third case, where $t_a < T$ but some later link in the cached route is broken, should be considered. However we discount this case since we assume the link broken is due to the mobility of MRs only. We assume that all links in the active route remain intact except for the first and last links. That is, the links between MRs and their neighbour nodes.

6.5.2 Routing Overhead in Multipath DSR

We assume n_S obtains multiple routes to n_D through a single route discovery process. The overhead is the number of packets in route discovery plus the route

reply packets from n_D to n_S. If all the routes to n_D are expired, a new route discovery process must be carried out. We assume that the route discovery process reveals σ paths through their neighbours, and that the number of hops from n_S to n_D is h_i, $2 \leq i \leq \sigma$. The overhead for flooding the route request is no more than $(n-1)$, plus the route replies along each of the σ paths, $\Sigma_{i=1}^{\sigma} h_i$.

Let t_a be the arrival time of a new packet destined for n_D, and T be the route expiry time for the largest surviving route. The overhead associated with the acquiring of a multipath between n_S and n_D is given by

$$\mathrm{OH} = \begin{cases} 0 & t_a < T \\ (n-1) + \Sigma_{i=1}^{\sigma} h_i & t_a > T. \end{cases} \tag{6.2}$$

The route expiry time should equal the time when the last route in the cache breaks, called the cache residual time. In order to calculate the expected overhead, we need to determine the cache residual time as well as the number of hops in the paths sending route replies in (6.1) and (6.2). We will look at each separately.

6.6 Time to Route Discovery

The biggest factor driving the network overhead is the initialisation of a route discovery process. Route failure only occurs due to n_S and n_D movement. We calculate the time until all of the routes in the cache fail (cache residual time). We compare the path residual time for single route, cache residual time for multiple cached routes through random neighbours, and cache residual time for multiple cached routes through selected neighbours as proposed in our multipath DSR. We demonstrate that the cache residual time is significantly longer when paths are selected as we recommend.

6.6.1 Travelling Distance to Link Failure

The MR/neighbour nodes scenario is illustrated in Fig. 6.3. To determine when the link between the MR and a given neighbour node, n_i, fails, we let n_i be at angle θ_i and distance $d_{o,i}$ from the MR. We let the angle between the direction of movement of the MR, θ_m, and its neighbour be $\delta_i = \theta_m - \theta_i$. Then the distance that the MR has to travel in direction θ_m, before the link with n_i fails, is given by

$$d_i = d_{o,i}\cos(\delta_i) + \sqrt{r^2 - d_{o,i}^2 \sin^2(\delta_i)}. \tag{6.3}$$

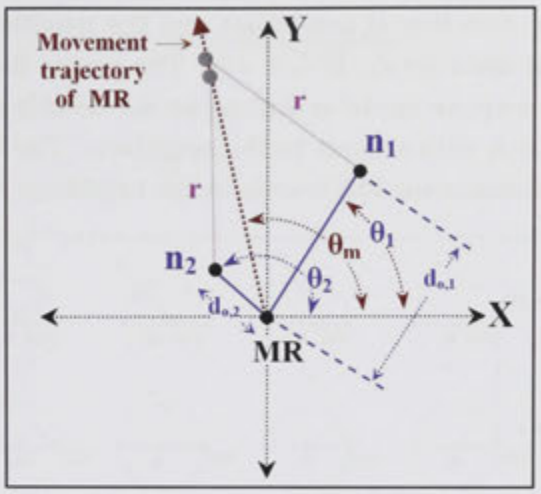


Figure 6.3: An example of two neighbour nodes, n_1 and n_2, at angles θ_1 and θ_2 and distance $d_{o,1}$ and $d_{o,2}$, respectively. The MR moves in a straight line in direction θ_m. Grey dots indicate MR location when the links to n_1 and n_2, respectively, fail.

6.6.2 Link Residual Time

The time from when the MR moves from its initial position to the point where the link with n_i fails is called the *link residual time*. The time required for the MR to move distance d_i at velocity v is given by

$$R_i = \frac{d_i}{v}. \tag{6.4}$$

Referring to Fig. 6.4, it can be seen that the link residual time decreases with $d_{o,i}$

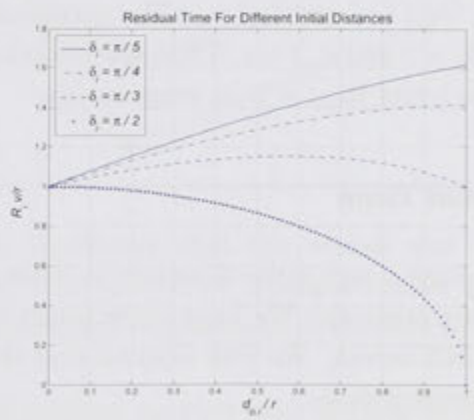


Figure 6.4: Normalized link residual time as it varies with the number of neighbour nodes and initial distance, between 0 and r from (6.3).

if the angle between direction of movement and the neighbour is $\delta_i = \pi/2$, and increases with initial distance $d_{o,i}$ if $\delta_i < \pi/3$. The reason for this can be seen in Fig. 6.5. Here, the 'reception circle' is centred on the neighbour node, and the MR angle of movement is δ_i with respect to the neighbour. The link residual time is proportional to the distance the MR travels in the neighbour's reception circle.

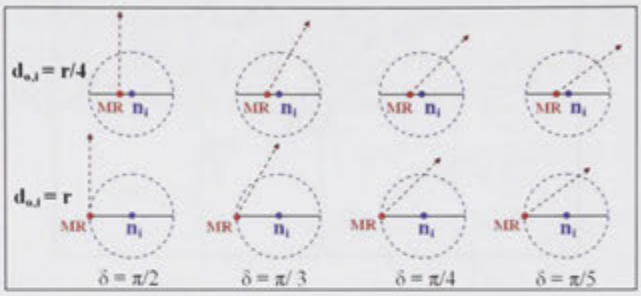


Figure 6.5: Circles have radius, r, centred on the nearest neighbour, n_i, which is located at distance $d_{o,i}$ from the MR. The MR moves in the direction of angle δ_i. The direction of movement is denoted by a dotted line.

Since the nodes are uniformly distributed within the sensing field, the probability density of distance between the MR and its neighbours linearly increases with the distance between them. The PDF can be described by $f(d_o) = 2d_o/r^2$ [110]. Assume that $\theta_i \sim U[0, 2\pi)$ and $\theta_m \sim U[0, 2\pi)$, then the link residual time is a random variable denoted R_{RN}. We derive the PDF and CDF from (6.3) and (6.4). The PDF and CDF of link residual time are $f_{R_{RN}}(t; v)$ and $F_{R_{RN}}(t; v)$, respectively, and are given in Appendix B.

Fig. 6.6 shows the PDF of R_{RN}. The minimum distance travelled until a link breaks is zero when $|\theta_m - \theta_i| = \pi$ and $d_o, i = r$. The maximum travelled distance is $d_{i,\max} = 2r$ when $|\theta_m - \theta_i| = 0$ and $d_o, i = r$. Thus, the minimum and maximum link residual time are $R_{\min} = 0$ and $R_{\max} = 2r/v$, respectively.

6.6.3 Cache Residual Time

The probability of initiating a new route discovery is determined by the topology scenario and the caching strategy. We have determined when the link to each neighbour fails as the MR moves. We also assume that the cache stores a path through different neighbour nodes.

Fig. 6.7 illustrates the scenario when the source MR caches multiple routes to the destination MR through its neighbour nodes. When n_S and n_D move at average velocity, v_S and v_D, respectively, the cache residual time depends on the

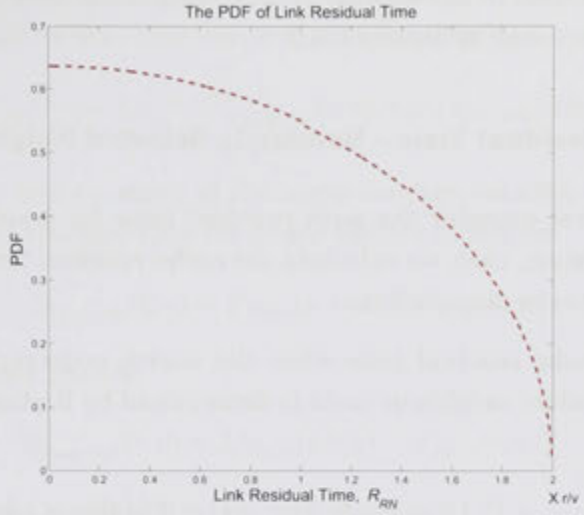


Figure 6.6: The PDF of link residual time between MR and its neighbour from from (B.1), (B.3) and (B.5) in Appendix B. The MR moves in random direction with v velocity.

links failure between n_S and its neighbours in addition to the link failure between n_D and its neighbours. We will calculate the distribution of the link residual time for the two ends in order to find the distribution of cache residual time.

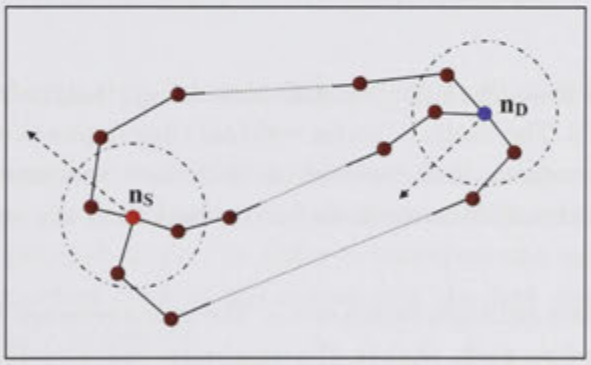


Figure 6.7: Example illustrates that the source and destination MRs move in random directions. The n_S cached multiple routes to n_D through different neighbour nodes. The direction of movement is denoted by a dotted arrow line.

The maximum link residual time between MR and its neighbour increases with the number of cached routes, since that decreases the angle between the neighbour node and direction of movement. First we consider the cache residual time when the route to destination is via randomly arranged neighbours. Then we consider

the cache residual time in our routing scheme where the first hop neighbours are θ-separated, as discussed in Section 6.4.2.

6.6.3.1 Cache Residual Time – Randomly Selected Neighbours

In this case we first consider the path residual time for a single route between source and destination, then we calculate the cache residual time when the source node caches σ routes to destination.

The PDF of cache residual time when the source node caches a single route, $R_{\text{path},RN}$, via a random neighbour node is determined by finding the CDF first, as follows.

$$\begin{aligned} F_{R_{\text{path},RN}}(t; v_S, v_D) &= Pr\{R_{S,i} < t \;\&\; R_{D,j} > t\} + Pr\{R_{S,i} > t \;\&\; R_{D,j} < t\} \\ &\quad + Pr\{R_{S,i} < t \;\&\; R_{D,j} < t\} \\ &= F_{R_{RN}}(t; v_S) - F_{R_{RN}}(t; v_S) F_{R_{RN}}(t; v_D) + F_{R_{RN}}(t; v_D), \end{aligned} \tag{6.5}$$

where $R_{S,i}$ and $R_{D,j}$, $1 < i, j \leq d$ are the link residual time between the n_S and n_D and their i-th and j-th neighbours, respectively. Taking the derivative of (6.5) it can be seen that the PDF of the path residual time is given by

$$\begin{aligned} &f_{R_{\text{path},RN}}(t; v_S, v_D) \\ &= f_{R_{RN}}(t; v_S) - F_{R_{RN}}(t; v_S) f_{R_{RN}}(t; v_D) - f_{R_{RN}}(t; v_S) F_{R_{RN}}(t; v_D) + f_{R_{RN}}(t; v_D), \end{aligned} \tag{6.6}$$

where $f_{R_{RN}}(t; v)$ is from (B.1), (B.3) and (B.5) and $F_{R_{RN}}(t; v)$ is from (B.2), (B.4) and (B.6) in Appendix B. The limits of cache residual time depends on the link residual time between the source and destination and their corresponding neighbours. Thus, the minimum and maximum cache residual time are equal to 0 and $2r/v$, respectively.

When n_S stores σ multiple routes to n_D. The cache residual time is equal to the time of the last broken path, that is, the maximum cache residual time. The PDF of the maximum residual time is determined by finding the CDF first, as follows

$$\begin{aligned} F_{R_{cache,RN}}(t; v_S, v_D) &= Pr\{\max\{R_i\} < t\} \\ &= Pr\{(R_1 < t) \cup (R_2 < t) \cup \cdots \cup (R_\sigma < t)\} \\ &= \prod_{i=1}^{\sigma} Pr\{R_i < t\} \\ &= F^{\sigma}_{R_{\text{path},RN}}(t; v_S, v_D), \end{aligned} \tag{6.7}$$

where R_i is the residual time of a single path. Taking the derivative of (6.7) it can be seen that the PDF of the cache residual time is given by

$$f_{R_{cache,RN}}(t; v_S, v_D) = \sigma F^{\sigma-1}_{R_{\text{path},RN}}(t; v_S, v_D) f_{R_{\text{path},RN}}(t; v_S, v_D). \tag{6.8}$$

Assume the n_S and n_D move at the same average velocity, $v_S = v_D = v$, then the CDF of the path residual time for single path is as follows

$$F_{R_{\text{path},RN}}(t; v) = F_{R_{RN}}(t; v)(2 - F_{R_{RN}}(t; v)), \tag{6.9}$$

and the PDF of single path is

$$f_{R_{\text{path},RN}}(t; v) = 2 f_{R_{RN}}(t; v)(1 - F_{R_{RN}}(t; v)). \tag{6.10}$$

The CDF of the cache residual time for multipath is

$$F_{R_{cache,RN}}(t; v) = F^{\sigma}_{R_{\text{path},RN}}(t; v), \tag{6.11}$$

and the PDF of the cache residual time for multipath is

$$\begin{aligned} f_{R_{cache,RN}}(t; v) &= \sigma F^{\sigma-1}_{R_{\text{path},RN}}(t; v) f_{R_{\text{path},RN}}(t; v) \\ &= 2\sigma f_{R_{RN}}(t; v)(1 - F_{R_{RN}}(t; v)) F^{\sigma-1}_{R_{RN}}(t; v)(2 - F_{R_{RN}}(t; v))^{\sigma-1}. \end{aligned} \tag{6.12}$$

Again, the minimum and maximum cache residual times are equal to 0 and $2r/v$, respectively.

6.6.3.2 Cache Residual Time – Our Routing Scheme

In our routing scheme, the source MR selects routes through neighbours that are closest to the direction of movement and destination MR selects routes through neighbours at θ-separated angle. In order to calculate the cache residual time, we consider link residual time at the source and the link residual time at the destination separately, as follows

- Link Residual Time at the Source, $R_{S,SN}$

  Assume that n_S stores multiple routes to n_D, each commencing with a different neighbour node n_i. We assume n_S selects the route to n_D through neighbours that are closest to the direction of movement. The residual time at the source is the maximum of all the residual times to individual neighbours. The PDF of the maximum link residual time $R_{S,SD}$ at the source

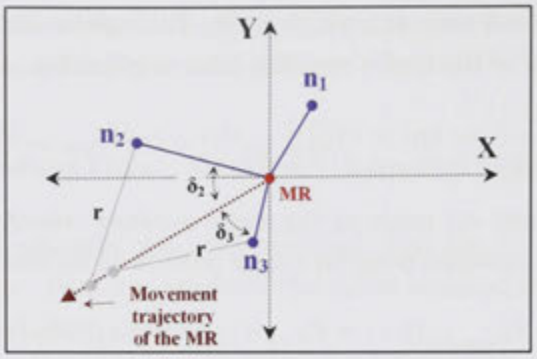


Figure 6.8: An example illustrates source MR cached routes to destination MR through appropriately distributed neighbour nodes of the destination, n_1, n_2 and n_3. Nodes n_2 and n_3 are two adjacent neighbours to MR at angles δ_2 and δ_3, respectively, from the direction of movement.

is determined by finding the CDF first, as follows.

$$\begin{aligned} F_{R_{S,SN}}(t; v_S) &= Pr\{\max\{R_i\} < t\} \\ &= Pr\{(R_1 < t) \cup (R_2 < t) \cup \cdots \cup (R_\sigma < t)\} \\ &= \prod_{i=1}^{\sigma} Pr\{R_i < t\} \\ &= F^{\sigma}_{R_{RN}}(t; v_S), \end{aligned} \tag{6.13}$$

where R_i is the link residual time of the link between the source node and its i-th neighbour. Taking the derivative of (6.13) it can be seen that the PDF of the maximum R_i value is given by

$$f_{R_{S,SN}}(t; v_S) = \sigma F^{\sigma-1}_{R_{RN}}(t; v_S) f_{R_{RN}}(t; v_S), \tag{6.14}$$

where $f_{R_{RN}}(t; v)$ is from (B.1), (B.3) and (B.5) and $F_{R_{RN}}(t; v)$ is from (B.2), (B.4) and (B.6) in Appendix B.

- Link Residual Time at the Destination, $R_{D,SN}$

  Now we consider the link residual time at the destination MR. Fig. 6.8 illustrates a destination MR selects routes through appropriately distributed neighbour nodes (neighbours θ-separated). In order to calculate the distribution of link residual time, we considered the worst-case scenario where n_D moves exactly between its two adjacent neighbours. The PDF and CDF of the individual link residual time for different values of neighbour separation angle, δ_i, are as follows.

  Case I: $\delta_i = \pi/2$ **(Two neighbours)**

  The link residual time, is equal to $R_{\max} = r/v_D$ and $R_{\min} = 0$ respectively

when $d_{o,i}$ is at its limiting values 0, r, as shown in Fig. 6.4. In this case $\sin(\delta_i) = 1$ and $\cos(\delta_i) = 0$ so (6.3) simplifies significantly, and the PDF and CDF of the link residual time are given by

$$f_{R_{SN}}(t; v_D) = \frac{2tv_D^2}{r^2}, \tag{6.15}$$

$$F_{R_{SN}}(t; v_D) = \frac{t^2 v_D^2}{r^2}. \tag{6.16}$$

Case II: $\delta_i = \pi/3$ **(Three neighbours)**

The minimum link residual time, is equal to $R_{\min} = r/v_D$ for $d_{o,i}$ equal to both 0 and r. The maximum link residual time $R_{\max} = r/(v_D \sin(\pi/3))$ is achieved when $d_{o,i} = r/\tan(\pi/3)$, as shown in Fig. 6.4. In this case $\sin(\delta_i) = \sqrt{3}/2$ and $\cos(\delta_i) = 0.5$, so the PDF and CDF of the link residual time are given by

$$f_{R_{SN}}(t; v_D) = \frac{3t^2 v_D^3 - 2r^2 v_D}{r^2\sqrt{r^2 - 0.75t^2 v_D^2}}, \tag{6.17}$$

$$F_{R_{SN}}(t; v_D) = 1 - \frac{2tv_D}{r^2}\sqrt{r^2 - 0.75t^2 v_D^2}. \tag{6.18}$$

Case III: $\delta_i \leq \pi/4$ **(Four or more neighbours)**

The minimum link residual time, is equal to $R_{\min} = r/v_D$ for $d_{o,i} = 0$. The maximum link residual time, $R_{\max} = 2r\cos(\delta)/v_D$, is achieved when $d_{o,i} = r$. Since δ_i varies, (6.3) must be treated as general. The PDF and CDF of the link residual time are given by

$$f_{R_{SN}}(t; v_D) = \frac{v_D}{r^2}(4tv_D\cos^2(\delta) - 2tv_D - \frac{2\cos(\delta)(2t^2v_D^2\cos^2(\delta) - 2t^2v_D^2 + r^2)}{\sqrt{t^2v_D^2\cos^2(\delta) - t^2v_D^2 + r^2}}), \tag{6.19}$$

$$F_{R_{SN}}(t; v_D) = \frac{1}{r^2}(r^2 - t^2v_D^2 + 2t^2v_D^2\cos^2(\delta) - 2tv_D\cos(\delta)\sqrt{t^2v_D^2\cos^2(\delta) - t^2v_D^2 + r^2}). \tag{6.20}$$

Each of the CDF functions is illustrated in Fig. 6.9.

The residual time at the destination, $R_{D,SN}$, is the largest of the residual times to the two neighbours that n_D passes between. Since the n_D selects routes that are θ-separated at the neighbour node, the last link to break is via the neighbour node that is in the closest position relative to destination

movement. In this scenario the PDF of the destination link residual time, $R_{D,SN}$, is determined by finding the CDF first, as follows:

$$\begin{aligned} F_{R_{D,SN}}(t) &= Pr\{\max\{R_{SN} < t\}\} \\ &= Pr\{(R_{SN1} < t) \cup (R_{SN2} < t)\} \\ &= F^2_{R_{SN}}(t; v_D), \end{aligned} \tag{6.21}$$

where R_1 and R_2 indicate residual times for the two closest neighbours. Taking the derivative of (6.21) it can be seen that the PDF of the maximum R_i value is given by

$$f_{R_{D,SN}}(t; v_D) = 2F_{R_{SN}}(t; v_D) f_{R_{SN}}(t; v_D). \tag{6.22}$$

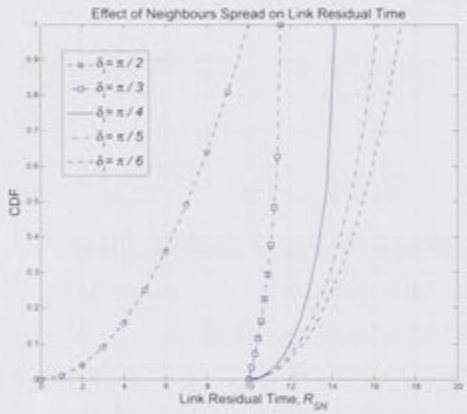


Figure 6.9: The CDF of the link residual time, for our scenario, where n_S moves between two neighbours from (6.16), (6.18) and (6.20). The angle between the direction of movement and neighbours varies from $\pi/6$ to $\pi/2$ and the node moves at velocity $v_D = 0.1r$.

Now, we can calculate the cache residual time in our routing scheme by calculating the CDF of cache residual time as follows

$$\begin{aligned} F_{R_{cache,SN}}(t; v_S, v_D) &= Pr\{R_{S,SN} < t \;\&\; R_{D,SN} > t\} + Pr\{R_{S,SN} > t \;\&\; R_{D,SN} < t\} \\ &\quad + Pr\{R_{S,SN} < t \;\&\; R_{D,SN} < t\} \\ &= F_{R_{S,SN}}(t; v_S)(1 - F_{R_{D,SN}}(t; v_D)) + (1 - F_{R_{S,SN}}(t; v_S)) F_{R_{D,SN}}(t; v_D) \\ &\quad + F_{R_{S,SN}}(t; v_S) F_{R_{D,SN}}(t; v_D) \\ &= F^{\sigma}_{R_{RN}}(t; v_S) + F^2_{R_{SN}}(t; v_D) - F^{\sigma}_{R_{RN}}(t; v_S) F^2_{R_{SN}}(t; v_D). \end{aligned} \tag{6.23}$$

Taking the derivative of (6.23) it can be seen that the PDF of the cache R_i value is given by

$$f_{R_{cache,SN}}(t; v_S, v_D) = \sigma F_{R_{RN}}^{\sigma-1}(t; v_S) f_{R_{RN}}(t; v_S)(1 - F_{R_{SN}}^{2}(t; v_D)) + 2F_{R_{SN}}(t; v_D) f_{R_{SN}}(t; v_D)(1 - F_{R_{RN}}^{\sigma}(t; v_S)). \quad (6.24)$$

Fig. 6.10 shows a comparison of the PDF of the cache residual time when the source MR caches random paths to destination with our routing scheme. The result shows that the cache residual time is larger in our routing scheme for the same number of paths. In addition, the cache residual time increases with the number of paths. It can be seen that the cache residual time in our scheme cannot reach the maximum cache residual time, $2r/v$, since we considered the worst-case scenario that the destination MR moves exactly between its two adjacent neighbours.

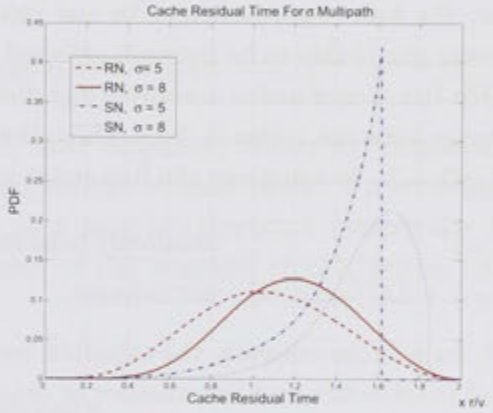


Figure 6.10: The PDF of the cache residual time, for randomly selected neighbours from (6.12) and our routing scheme from (6.24) when the number of paths equals to 5 and 8 and $v = v_S = v_D$.

6.6.4 Packet Arrival Time

Routes are only sought if n_S actually needs a route to send a packet from n_S to n_D. Packet arrival time, t_a, is generally modelled as having an exponential distribution, with parameter λ_a appropriate to the given network. We use this model here. The PDF and CDF of arrival times are given by

$$f_a(t) = \lambda_a e^{-\lambda_a t}, \quad (6.25)$$

$$F_a(t) = 1 - e^{-\lambda_a t}. \quad (6.26)$$

In this section we have presented a statistical model of the topology scenario. In Section 6.8 we will use this model to develop an analytical model of the expected overhead. First we estimate the hop count between the source and destination that is necessary to calculate the expected overhead.

6.7 Path Length in Hops

In this section we calculate the expected hop count between the source and destination MR. We use our model demonstrated in Chapter 5 to estimate the hop count assuming the distance between the source and destination is L_{SD}. Since the neighbour nodes are independent and identically distributed in the sensing field, the expected distance between the source and destination is L_{SD} when neighbour nodes are randomly selected. Thus, the expected hop count is $E\{hop|L_{SD}\}$. However, the hop count is larger in our routing scheme since the destination MR augments the routes to be through selected neighbours. The work in [110] shows that when the sensor nodes are uniformly distributed in the sensing field, the average distance between nodes is $2r/3$. Thus, the number of extra hops in our scheme is $2E\{hop|2r/3\}$. In summary the hop count is given by

$$E\{\mathrm{h}\} = \begin{cases} E\{hop|L_{SD}\} & \text{randomly selected neighbours,} \\ E\{hop|L_{SD}\} + 2E\{hop|2r/3\} & \text{our scheme.} \end{cases} \tag{6.27}$$

Referring to Chapter 5, for a dense network, the expected hop count equals $L_{SD}/r + 1$ for randomly selected neighbours and the number of extra hops equals $10/3$ for our routing scheme.

6.8 Expected Overhead

In this section we consider the expected overhead with the occurrence of data-packet requests and new route requests. The expected overhead is equal to the cost of an individual route discovery process, multiplied by the probability that the route is necessary. The probability of route discovery is determined by the topology scenario and the caching strategy. We start with the expected overhead in single path routing, then consider the overhead in multipath routing.

6.8.1 Single Path

In this case we assume that the cache at n_S stores a single route for each n_D. In order to incur routing overhead, the route must have broken prior to the arrival

of the next packet to send. That is, $t_a > R$. As t_a and R are independent random variables, the probability that $t_a > R$ is

$$\begin{aligned} Pr\{t_a > R\} &= \int_{R_{\min}}^{R_{\max}} \int_{t}^{\infty} f_a(t_a) f_R(t) dt_a dt \\ &= \int_{R_{\min}}^{R_{\max}} (1 - F_a(t)) f_R(t) dt \\ &= \int_{R_{\min}}^{R_{\max}} e^{-\lambda_a t} f_R(t) dt, \end{aligned} \tag{6.28}$$

where R is the residual time, and represents $R_{path,RN}$ in which case $f_R(t)$ will be given by (6.10). Similarly R may be the cache residual time $R_{cache,RN}$ or $R_{cache,SN}$.

Assume that n_S and n_D move at the same average velocity, $v = v_S = v_D$. The corresponding values of R_i are $R_{\min} = 0$ and $R_{\max} = 2r/v$. Then, the expected value of the routing overhead is

$$E\{\text{OH}|\text{single path}\} = (n + E\{\text{h}\} - 1) \int_0^{2r/v} e^{-\lambda_a t} f_{R_{\text{path},RN}}(t) dt. \tag{6.29}$$

We have used the fact that neighbour nodes are independently placed, so link residual times are independent of each other. Unfortunately, there is no closed form solution to (6.29), so it must be calculated numerically. Note that the expected overhead is independent of the number of neighbours when only one route is cached.

6.8.2 Passive Multipath

We assume that the routing protocol is configured to avoid route discovery as long as possible. If there are multiple routes in the cache, the moving source node will progressively lose connection with the first hop in each route. If there are routes originating with each neighbour node, the last route to be broken in this way will correspond to the closest neighbour in the direction of movement.

The cache at n_S stores multiple routes to n_D, via different neighbour node. Assume a route to n_D is stored with probability P. The probability that there are $\sigma \leq d$ routes cached is equal to

$$\Pr(\sigma) = \frac{d!}{\sigma!(d-\sigma)!} P^{\sigma} (1-P)^{d-\sigma}. \tag{6.30}$$

In particular, if $P = 1$, then there are $\sigma = d$ cached routes with probability 1.

As mentioned above, in this case the source node can use the route through the neighbour closest to its direction of movement, θ_m, so the cache residual time

is determined by (6.8). No route discovery process is incurred until the link to that closest neighbour breaks.

Combining (6.8)(6.28) and (6.30), we can determine an expression for expected overhead when multiple paths are cached.

$$
\begin{aligned}
&E\{\text{OH}|\text{passive multipath}\} \\
&= (n + E\{\text{h}\} - 1) \sum_{\sigma=0}^{d} \int_{R_{\min}}^{R_{\max}} Pr(\sigma) \cdot e^{-\lambda_a t} \cdot f_{R_{cache,RN}}(t; v_S, v_D) dt \\
&= (n + E\{\text{h}\} - 1) \Big[(1-P)^d + \sum_{\sigma=1}^{d} \frac{d!}{\sigma!(d-\sigma)!} P^\sigma (1-P)^{d-\sigma} \\
&\quad \cdot \int_0^{2r/v} e^{-\lambda_a t} f_{R_{cache,RN}}(t; v_S, v_D) dt \Big]. \qquad (6.31)
\end{aligned}
$$

Assume that the n_S and n_D move at the same average velocity, v, then the expected overhead is as follows

$$
\begin{aligned}
&E\{\text{OH}|\text{passive multipath}\} \\
&= (n + E\{\text{h}\} - 1) \sum_{\sigma=0}^{d} \int_{R_{\min}}^{R_{\max}} Pr(\sigma) \cdot e^{-\lambda_a t} \cdot f_{R_{cache,RN}}(t; v) dt \\
&= (n + E\{\text{h}\} - 1) \Big[(1-P)^d + \sum_{\sigma=1}^{d} \frac{d!}{\sigma!(d-\sigma)!} P^\sigma (1-P)^{d-\sigma} \\
&\quad \cdot 2\sigma \int_0^{2r/v} e^{-\lambda_a t} f_{R_{RN}}(t; v)(1 - F_{R_{RN}}(t; v)) F_{R_{RN}}^{\sigma-1}(t; v)(2 - F_{R_{RN}}(t; v))^{\sigma-1} dt \Big], \\
& \qquad (6.32)
\end{aligned}
$$

where $F_{R_{cache,RN}}(t; v)$ is from (6.12). In the special case where $P = 1$, then $\sigma = d$. That is, n_S stores one route to n_D through different neighbour node. We find that the expected overhead when there are d paths in the cache is

$$
\begin{aligned}
&E\{\text{OH}|\text{passive multipath}\} \\
&= (n + E\{\text{h}\} - 1) \int_{R_{\min}}^{R_{\max}} e^{-\lambda_a t} F_{R_{cache,RN}}(t; v) dt \\
&= (n + E\{\text{h}\} - 1) 2\sigma \int_0^{2r/v} e^{-\lambda_a t} f_{R_{RN}}(t; v)(1 - F_{R_{RN}}(t; v)) F_{R_{RN}}^{d-1}(t; v)(2 - F_{R_{RN}}(t; v))^{d-1} dt \\
& \qquad (6.33)
\end{aligned}
$$

There is no closed form solution to either (6.32) or (6.33), so they must be calculated numerically. Note that in both cases the expected overhead is now dependent on the number of neighbours.

6.8.3 Active Multipath

In this section we consider the expected overhead for active multipath. First we calculate the expected overhead when the source and destination randomly select paths through their neighbours, then we consider the overhead in our routing scheme as described in Section 6.4.

In active multipath, we consider the expected overhead incurred by data-packet requests and new route requests. The expected overhead is equal to the cost of an individual route discovery process, multiplied by the probability that the route is broken.

In order to find the expected overhead when the n_S and n_D randomly select paths, we combining equations (6.2) and (6.8) . We have the following expression for the expected overhead when n_S and n_D move at velocity v_S and v_D, respectively.

$$
\begin{aligned}
&E\{\text{OH}|\text{active multipath, randomly selected neighbours}\} \\
&= (n + \sigma E\{h\} - 1) \int_0^{2r/v} e^{-\lambda_a t} f_{R_{cache,RN}}(t; v_S, v_D) dt \\
&= (n + \sigma E\{h\} - 1)\sigma \int_0^{2r/v} e^{-\lambda_a t} F^{\sigma-1}_{R_{\text{path},RN}}(t; v_S, v_D) f_{R_{\text{path},RN}}(t; v_S, v_D) dt, \qquad (6.34)
\end{aligned}
$$

and the expected overhead in our routing scheme is

$$
\begin{aligned}
&E\{\text{OH}|\text{active multipath, path through } \theta\text{-separated angle}\} \\
&= (n + \sigma E\{h\} - 1) \int_{R_{\min}}^{R_{\max}} e^{-\lambda_a t} f_{R_{path,SN}}(t; v_S, v_D) dt. \qquad (6.35)
\end{aligned}
$$

Assume that the n_S and n_D move at the same average velocity, v, then the expected overhead is a follows

$$
\begin{aligned}
&E\{\text{OH}|\text{active multipath, randomly selected neighbours}\} \\
&= (n + \sigma E\{h\} - 1) 2\sigma \int_0^{2r/v} e^{-\lambda_a t} f_{R_{RN}}(t; v)(1 - F_{R_{RN}}(t; v)) \\
&\qquad F^{\sigma-1}_{R_{RN}}(t; v)(2 - F_{R_{RN}}(t; v))^{\sigma-1} dt, \qquad (6.36)
\end{aligned}
$$

$$
\begin{aligned}
&E\{\text{OH}|\text{active multipath, path through } \theta\text{-separated angle}\} \\
&= (n + \sigma E\{h\} - 1) \int_{R_{\min}}^{R_{\max}} e^{-\lambda_a t} (\sigma F^{\sigma-1}_{R_{RN}}(t; v) f_{R_{RN}}(t; v)(1 - F^2_{R_{SN}}(t; v)) \\
&\qquad + 2F_{R_{SN}}(t; v) f_{R_{SN}}(t; v)(1 - F^{\sigma}_{R_{RN}}(t; v))) dt. \qquad (6.37)
\end{aligned}
$$

Recall that $f_{R_{RN}}(t; v)$ and $F_{R_{RN}}(t; v)$ are the PDF and CDF obtained in Appendix (B) and $f_{R_{SN}}(t; v)$ and $F_{R_{SN}}(t; v)$ are the PDF and CDF obtained in Section 6.6.3.2. The values of $R_{\min}$ and $R_{\max}$ depend on the number of paths, σ. There is no closed form solution to (6.36) and (6.37), so they must be calculated numerically.

6.9 Results

In this section we present and discuss results of theoretical calculations and Monte-Carlo simulations, conducted in MATLAB, for expected overhead given in (6.32), (6.36) and (6.37).

In the simulation, nodes are uniformly distributed in a circular sensing field with radius $6r$. The maximum transmission range (r) of each node is set to $100m$. The source MR sends data-packets to the destination MR at arrival time t_a seconds while they move in a random direction and average velocity, v.

Since the MR could be carried by a variety of mobile entities such as a robot or Unmanned Aerial Vehicle (UAV) plane, we consider different velocities for the MRs, such as 50, 10 and 5 m/s ($0.5r$, $0.1r$ and $0.05r$ m/s), which we will refer to as high, medium and low velocity, respectively.

For each instance of the simulation, sensor nodes are uniformly distributed in the sensing field and the network connectivity is tested to ensure there are no partition subnetworks. To isolate the effect of multipath routes, the source and destination MRs are randomly deployed in the sensing field such that the initial distance between them is L_{SD}. Each MR caches the location of its neighbours in the neighbour table. The packet arrival time, t_a, at source MR, is then generated according to an exponential distribution with parameter $\lambda_a = 0.1$, which will be compared to the cache residual time, and the overhead is counted. In order to find the number of hops between the neighbour nodes and the destination, the BFS algorithm is used to find a routing tree rooted at the destination MR.

6.9.1 Low Node Density Network – Randomly Chosen Paths

First, we consider a network with low node density, $d = 6$. We consider the scenario of passive and active multipath routing. The expected overhead for passive and active multipath routing as they vary with number of paths and MRs velocities are illustrated in Fig. 6.11. We assume the distance between the source and destination MRs is $L_{SD} = 6r$. In passive multipath, we consider different values for probability, P, for the source MR to cache multipath routes. That is, for any particular number of multipaths, σ, we choose P such that the expected number of paths, $E\{\sigma\}$ equals σ. Thus, for the expected number of paths equals $2, 3, 4, 5$, and 6, the probability, P, is $1/3, 1/2, 2/3, 5/6$ and 1, respectively. The theoretical overhead for passive and active routing are from (6.32) and (6.36), respectively, and they match well with the corresponding simulation overhead.

The results shows that the number of paths and velocities, have a significant

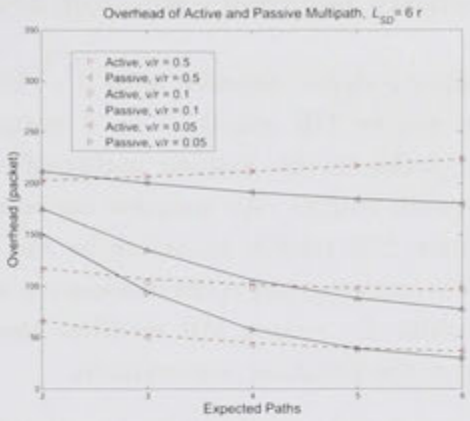


Figure 6.11: Overhead as it varies with source and destination MRs velocities, for passive and active multipath, (6.32) and (6.36), respectively. $L_{SD} = 6r$ and $d = 6$. Simulation results are depicted by markers while theoretical results are depicted by lines.

effect on overhead. For passive multipath routing, the overhead decreases with the number of paths for different MR velocities, since increase in the number of paths decreases the number of route discoveries by providing alternative routes to the destination. The high overhead for the passive routing when the probability of getting routes is small is because there is significant probability that no path actually exists in the cache, even though the expected number of paths is 2 or 3. However, the effect of the number of paths on overhead is less pronounced at high velocity, since at high velocity the path residual time for all routes is very small so that it has no effect on overhead. In practice the number of paths in passive multipath is determined by the probability, P, of learning paths, which is determined by the amount of traffic in the network.

Comparing the overhead for passive and active multipath routing, the results show that at low and medium velocities, the overhead is lower for the active multipath routing than the passive one for a small number of paths. However, at high velocity, the overhead increases with the number of paths for active multipath routing since, as the velocity increases, the number of route discoveries increases which outweighs the reduction in overhead due to providing alternative paths to destination.

6.9.2 High Node Density Network – Random or Selected Paths

In this section we consider a dense network with $d = 20$ and active multipath routing scenario where source MR stores $\sigma < d$ routes to each destination because the number of paths in the passive multipath cannot be guaranteed (since it depends on network traffic). We consider our routing scheme such that the source and destination MR modify routes to be via appropriately selected neighbours. Algorithm 2 is implemented at the destination MR to select neighbours at θ-separated angle, while the source MR modifies the routes to be via the neighbour that is closest to the direction of movement.

Fig. 6.12 shows the simulation and theoretical overhead for randomly selected multipath routing and our multipath routing scheme, (6.36) and (6.37) respectively, at low, medium and high velocities. It can be seen that the theoretical and simulation results match very well except that the theoretical overhead in our routing scheme is higher than that simulated when the number of paths equals 2 and 3. This is because we have considered the worst-case scenario in the calculation of the expected overhead in which the destination MR moves exactly between its adjacent neighbours (refer to Fig. 6.8). As the number of paths increases the difference in overhead between the theory and simulation decreases, as the angle between neighbours decreases. The results also show that in low velocity the overhead decreases with the number of paths. However, the reduction in the overhead decreases with the number of paths to be almost negligible after six paths.

To evaluate the performance of our routing scheme, we compare the routing overhead for our multipath routing scheme, randomly selected multipath routing and the Braided multipath routing scheme that is proposed in [11]. The Braided multipath routing scheme relaxed the requirement for node disjointedness and the alternate paths are partially disjoint from the primary path (the path with minimum hop count). It can be seen from Fig. 6.13 that under different velocities for the source and destination MRs, the routing overhead in our scheme is lower than for a randomly selected neighbour and the Braided multipath routing scheme. This is because the cache residual time in our routing scheme is higher than the cache residual time at randomly selected neighbours as shown in Fig. 6.10.

Moreover, it is interesting to see that the routing overhead for randomly selected neighbours is lower than that in the Braided multipath routing scheme. Since most of the neighbour nodes (of source and destination MR) of the alternative paths in the Braided scheme are very close to the neighbours of the primary path and hence the paths fail very quickly when the MRs move in the opposite direction to the neighbour nodes.

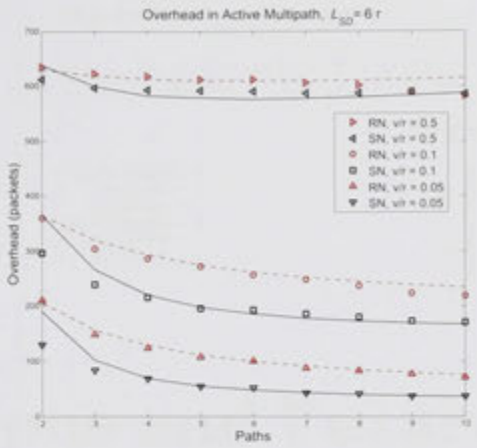


Figure 6.12: Overhead in a dense network as it varies with source and destination MRs velocities, for randomly selected paths and our multipath routing scheme, (6.36) and (6.37), respectively. $L_{SD} = 6r$ and $d = 20$. Simulation results are depicted by markers while theoretical results are depicted by lines.

The results also show that at high velocity and for a low number of paths the overhead decreases with the number of paths, since alternative routes to the destination decrease the number of route discovery processes. However, after a certain number of paths, the reduction in overhead, due to providing alternative paths, is outweighed by the route replies for each discovered path so the expected overhead increases. We see from the results that the minimum overhead can be achieved in our routing scheme when the source MR has six multipath routes.

6.10 Conclusion

In this chapter we considered multipath routing to provide an interaction between MRs. Sensor nodes are used to provide routing paths between the source and destination MRs via multihop communication. The number of paths in a multipath routing protocol and the distribution of neighbouring nodes in the discovered routes are studied when the source and destination MRs move in arbitrary directions with various velocities. We considered passive and active multipath routing mechanisms. We proposed an active multipath routing scheme for a dense network where the source and destination MR select routes through appropriately selected neighbours. An analytical model has been developed and verified by simulation.

We show that the number of paths has a significate effect on the network

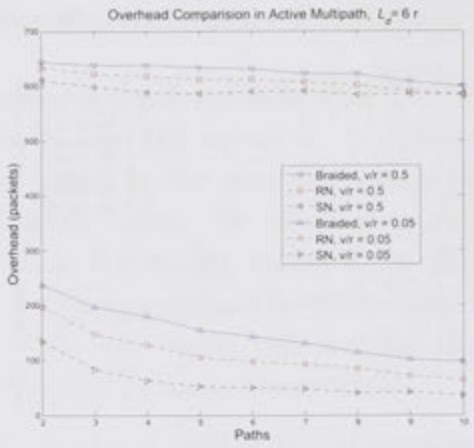


Figure 6.13: Overhead comparison in a dense network as it varies with source and destination MRs velocities, our multipath routing scheme, for randomly selected paths and the Braided scheme [11]. $L_{SD} = 6r$ and $d = 20$.

overhead for active and passive multipath routing. We show that at high velocity and in a low-node density network the number of paths has less effect for passive multipath while the overhead increases with the number of paths for active multipath routing.

We show that our routing scheme improves the network overhead compared with randomly selected neighbours and the Braided multipath routing scheme. In addition the network overhead is dependent on the MRs velocity and the number of paths corresponding to the neighbours of the source and destination MRs. For low and medium velocity, the network overhead decreases with the number of paths since that decreases the number of route discoveries by providing alternative routes to the destination. For high velocity when the number of paths is too small, the overhead is increased because of increased need for route discovery, otherwise, the overhead is increased because of an increased number of route reply packets. Thus the multipath routing can significantly reduce the network overhead. We have shown that, in most cases, the network overhead is minimized when the moving source and destination MR has six paths via appropriately distributed neighbours.

Chapter 7

Conclusions and Future Work

In this chapter, we state the conclusions drawn from the thesis. We also describe possible areas of future research arising from this work.

7.1 Conclusions

With the development of wireless sensor network technology and applications such as those that monitor the physical environment, there is increased focus on the use of mobile sensors to achieve desired network requirements. This thesis is concerned with exploring how the use of mobility in WSNs effects network performance. In WSNs the energy consumption of sensor nodes is a crucial consideration, since sensor nodes are powered by small batteries. It has been shown in Chapter 2 that using a mobile base station (BS) or mobile relays (MRs) can prolong network lifetime.

In this thesis we propose path planning for a mobile BS considering the time required for data gathering. We show the effect of BS mobility and its limitations. We also consider the use of MRs for data gathering. We highlight the interaction among MRs while they move in the sensing field for data gathering. We consider the effect of MR speed on path failure and how to design a routing protocol that decreases the effect of path failure among MRs due to mobility. Based on the above aspects, we draw the following conclusions.

(i) Using a mobile BS for data gathering improves network lifetime by balancing energy consumption among sensor nodes. Increased data-gathering delay is one of the disadvantages of this approach, since the speed of a mobile BS is very slow compared with the speed of data-packet travel in a multi-hop forwarding approach. In Chapter 3 we presented a cluster-based algorithm to

determine the trajectory of a mobile BS for data gathering within a specified delay time. Sensor nodes send their data to the cluster heads and the mobile BS roams the sensing field and visits only the cluster heads to gather sensing data. Our technique aims for an equal number of sensors in each cluster in order to achieve load balance among the cluster heads, since each cluster head has to forward the sensing data of the cluster nodes to the mobile BS. We show that there is a tradeoff between data-gathering delay and balancing energy consumption among sensor nodes.

(ii) The number of clusters is an important parameter that determines the amount of data each cluster head has to forward to the BS. We analyse our algorithm in Chapter 4 to show how to choose the number of clusters to ensure there is no packet loss as the BS moves between clusters. We provide an analytical solution to the problem in terms of the speed of the mobile BS. Simulation is performed to evaluate the performance of the proposed algorithm against the static case and to evaluate the distribution of energy consumption among the cluster heads. We show that the use of clustering with a mobile BS can improve the network lifetime and our proposed algorithm balances energy consumption among cluster heads.

(iii) There are significant constraints on the velocity of the mobile BS. Increasing the velocity of the mobile BS can help to reduce data-gathering delay, however, that increases the cost of the mobile BS and reduces the time available for data gathering from cluster heads. Moreover, in Chapter 4 we have shown that, for any given network, there is a maximum speed beyond which there will inevitably be packet loss.

(iv) As the network scale expands, the number of sensor nodes increases. In order for the cluster heads to get enough time to send cluster-sensing data to the BS, we need to increase the number of clusters. However, in the analysis of our algorithm in Chapter 4, we show that increasing the network scale causes packet losses even if the network nodes are clustered with the maximum number of clusters. For given network parameters, after a certain network radius, the time required for sending cluster-sensing data exceeds the residual time for data transfer from cluster head to the BS.

(v) In order to address the challenge of large-sensor networks, we consider the use of cooperating MRs for data gathering, in order to achieve no packet loss as network scale increases. To allow MRs to freely move in the sensing field, we assume the scenario that MRs can interact with each other using sensor nodes via multi-hop communication. A routing scheme to provide multipath routing among MRs with minimum network overhead is proposed

in Chapter 6.

(vi) As the MRs move in the sensing field for data gathering, paths among them fail as the links with their neighbours are broken. As the velocity of MRs increases, the frequency of path failure increases, which incurs more network overhead for finding new paths. In Chapter 6 we model the expected network overhead when the source MR caches multiple paths to the destination. We consider the effect of the number of paths in passive and active multipath routing for different MR speeds. In general increasing the number of paths improves network overhead. However, this is not the case for high MR speed in the active multipath case.

(vii) The effect of the distribution of neighbours of MRs is also studied in Chapter 6. Our active routing scheme allows the destination MR to create routes to the source via θ-separated neighbours. In spite of our routing scheme increasing the hop count between the source and destination, we show that the network overhead decreases compared with randomly selected multipath routing. This is because providing paths via neighbours in different directions increases the cache residual time between the source and destination MRs.

(viii) Common approaches to model hop count do not correctly address how the hop count is effected by low node density. Modeling the hop count between the source and destination MRs is required for the analysis of expected network overhead in Chapter 6. In Chapter 5 we propose an analytical model to estimate the hop count between source-destination pairs for arbitrary node density when network nodes are uniformly distributed in the sensing field. The effect of hop progress and connectivity on hop count are considered at different node densities. We consider the shortest path with the minimum number of hops between the source and destination. The analytical model is verified by simulation. We show that the hop count is small for small node degree and increases with increasing node degree to reach its maximum value when the node degree equals five. We refer to this network density as the threshold density. The hop count then decreases with increasing node degree to reach its minimum value. This model can be used in many applications such as localization, communication-protocol design and other areas.

7.2 Future Work

Research into the use of mobility in WSNs is yet to fully mature. More research is required to investigate the benefits that can be gained by using a mobile BS or MRs. In addition, the consequences of mobility on other aspects of network performance

should be analysed. There are a number of further research directions arising from the work presented in this thesis, including:

(i) The extension of our BS algorithm in Chapter 3 to consider data aggregation at the cluster heads in order to reduce the volume of sensing data. In this case, the neighbour nodes of cluster heads, rather than the cluster heads themselves, consume more energy than any other nodes in the network, since they have to relay the sensing data received from other sensor nodes to the cluster heads. Thus, changing the length of the BS tour changes the load balance among neighbour nodes of cluster heads in this case. We expect that the maximum velocity of the mobile BS (such that there are no packet losses) is higher than for our algorithm since the cluster heads have smaller volumes of data that need to be sent to the BS. Moreover, the network scale can also expand to be able to collect sensing data of more cluster-sensor nodes. The analysis of this scenario has to consider different types of data-aggregation functions and the delay associated with data aggregation.

(ii) The work in Chapter 3 can be extended to allow the BS to dynamically select cluster heads after a number of data-gathering rounds, based on the residual energy of cluster heads. In order to satisfy the delay requirement for data gathering and balance the energy consumption within each cluster, we assume the sensor nodes that are not currently cluster heads are candidates for being new cluster heads. A set of cluster heads is selected that are closest to the previous cluster head, such that the tour length is no greater than the maximum allowed tour length.

(iii) Connectivity and energy efficiency are the most fundamental issues for WSNs. Various topology-control algorithms have been proposed to maintain the connectivity of the communication graph and reduce the energy consumption of node transceivers, via adjusting transmission ranges. However, the effect of node mobility is not fully considered in the design of topology-control algorithms. In particular, the results of Fig. 6.4 shows that longer links give increased link residual time when these are sufficient well distributed neighbours. While longer links required more energy at the node, this will reduce total network energy consumption by avoiding control packet overhead incurred requesting a new route. For a given average node mobility, there will be an optimal transmission range so that links are sufficiently long to avoid unnecessary network overhead.

(iv) The effect of the wireless channel, such as path loss and interference can be incorporated. The selection of cluster heads in Chapter 3 can include cross-layer consideration by modifying the metric for selection of real cluster heads. In

addition to considering the distance to the virtual cluster head, the reliability of communication can be included. Channel effect can also be considered in the routing scheme for mobile relays in Chapter 6. The neighbour selection at the source has to select the neighbour that shows the highest communication reliability even if it is not the closest one to the direction of movement. In the same way the neighbour selection Algorithm 2 can be modified to consider the set of neighbours with maximum communication reliability. Therefore, the number and the distribution of neighbours and hence routing paths will depend on the communication environment.

Determining the location of sensor nodes is important in many aspects in WSNs. There is increased focus on the use of mobile BS for localization. The mobile BS (such as an UAV helicopter) can be used to estimate the location of limited energy and limited mobility sensor nodes, in applications such as animal tracking, where it is not possible to equip a GPS for each sensor node. Instead we assume only that the BS is equipped with a GPS, and aim to find the moving trajectory of a BS in order to localize as many sensors as possible with minimum time delay. We consider two mechanisms of sensor localization based on the ability of sensor nodes to use single hop or multi-hop communication:

(v) If sensor nodes are only capable of single hop communication, we assume sensor nodes are carried by small animals (such as a frog). Each sensor node periodically transmits a beacon signal at a specified frequency. The BS initially has to randomly roam the sensor field. When it receives a beacon from a sensor it estimates sensor location using, for example, a directional antenna. The estimated location of sensor nodes is buffered and used to dynamically modify the trajectory of the BS. The BS needs to be as close as possible to each sensor node to improve estimation accuracy. In addition, it has to cover most of the sensing field in order to increase the probability of visiting the transmission range of all sensor nodes.

(vi) If sensor nodes are capable of multi-hop communication, the BS periodically broadcasts a beacon packet to inform sensors to send a reply. Each sensor node receives the beacon packet, adds its address and rebroadcasts the packet. A reply packet is sent by each node to the BS through multi-hop routing, using the same path by which the beacon packet was received. When the BS receives the reply packet it estimates the distance of sensor nodes by using the number of hops and node density as illustrated in Chapter 5. In order to improve estimation accuracy and reduce delay the distance (maximum number of hops) between sensor nodes and the BS needs to to be minimized by planning the moving trajectory of the BS. An algorithm is needed to find the

trajectory of the BS that balances the hop count between the BS and sensor nodes within a specific time.

Appendix

A

The PDF and CDF of $L_{r,i}$

A.1 List of the PDF and CDF of the Remaining Distance to the Destination, $L_{r,i}$

We derive the PDF and CDF of $L_{r,i}$ from (5.1). Recall that $L_{r,i}$ is a function of two random variables $\phi = (\theta_d - \theta_i) \sim U(0, \pi)$ and $d_{o,i}$ is linear distributed with the distribution $f(d_o) = 2d_o/r^2$. Let $b = \frac{L_{SD}^2 + r^2 - l_r^2}{2L_o r}$ and $q(x) = \sqrt{l_r^2 - L_{SD}^2 \sin^2(x)}$ then the PDF and CDF of $L_{r,i}$ are given by

Case I: $L_{SD} - r < l_r < \sqrt{L_{SD}^2 - r^2}$

$$f_{L_r}(l_r) = \frac{l_r}{\pi r L_{SD}\sqrt{1-b^2}} - \frac{l_r}{\pi r^2}\left(\frac{L_{SD}b - q(\cos^{-1}(b))}{L_{SD}r\sqrt{1-b^2}} - 2\int_0^{\cos^{-1}(b)} \frac{L_{SD}\cos(\phi) - q(\phi)}{q(\phi)} d\phi\right) \tag{A.1}$$

$$F_{L_r}(l_r) = \frac{1}{r^2\pi}\int_0^{\cos^{-1}(b)} r^2 - \left(L_{SD}\cos(\phi) - q(\phi)\right)^2 d\phi \tag{A.2}$$

Case II: $\sqrt{L_{SD}^2 - r^2} < l_r \leq L_{SD}$

$$
\begin{aligned}
f_{L_r}(l_r) = & \frac{1}{\pi L_{SD}\sqrt{1-(\frac{l_r}{L_{SD}})^2}} - \frac{1}{r^2\pi}\Big(\frac{(L_{SD}\cos(\sin^{-1}\frac{l_r}{L_{SD}}) - q(\sin^{-1}(\frac{l_r}{L_{SD}})))^2}{L_{SD}\sqrt{1-(\frac{l_r}{L_{SD}})^2}} \\
& - \int_0^{\sin^{-1}(\frac{l_r}{L_{SD}})} \frac{2l_r(L_{SD}\cos(\phi)) - q(\phi)}{q(\phi)} d\phi \\
& - \frac{1}{\pi L_{SD}\sqrt{1-(\frac{l_r}{L_{SD}})^2}} + \frac{l_r}{\pi L_{SD} r\sqrt{1-b^2}} \\
& + \frac{1}{r^2\pi}\Big(\frac{(L_{SD}\cos(\sin^{-1}(\frac{l_r}{L_{SD}})) + q(\sin^{-1}\frac{l_r}{L_{SD}}))^2}{L_{SD}\sqrt{1-(\frac{l_r}{L_{SD}})^2}} - \frac{l_r(L_{SD}b + q(\cos^{-1}b))^2}{L_{SD}r\sqrt{1-b^2}} \\
& + \int_{\cos^{-1}(b)}^{\sin^{-1}(\frac{l_r}{L_{SD}})} \frac{2l_r(L_{SD}\cos(\phi) + q(\phi))}{q(\phi)} d\phi\Big)
\end{aligned} \tag{A.3}
$$

$$
\begin{aligned}
F_{L_r}(l_r) = & \frac{1}{r^2\pi}\int_0^{\sin^{-1}(\frac{l_r}{L_{SD}})} r^2 - (L_{SD}\cos(\phi) - q(\phi))^2\, d\phi \\
& - \frac{1}{r^2\pi}\int_{\cos^{-1}(b)}^{\sin^{-1}(\frac{li}{L_{SD}})} r^2 - (L_{SD}\cos(\phi) + q(\phi))^2\, d\phi
\end{aligned} \tag{A.4}
$$

Case III: $L_{SD} < l_r < L_{SD} + r$

$$
\begin{aligned}
f_{L_r}(l_r) = & \frac{1}{r^2\pi}\Big(\frac{-l_r(L_{SD}b + q(\cos^{-1}(b)))^2}{L_{SD}r\sqrt{1-b^2}} \\
& + \int_{\cos^{-1}(b)}^{\pi} \frac{2l_r(L_{SD}\cos(\phi) + q(\phi))}{q(\phi)} d\phi\Big) + \frac{l_r}{\pi L_{SD}r\sqrt{1-b^2}}
\end{aligned} \tag{A.5}
$$

$$
F_{L_r}(l_r) = \frac{1}{r^2\pi}\int_{\cos^{-1}(b)}^{\pi} (L_{SD}\cos(\phi) + q(\phi)d\phi)^2 + \frac{1}{\pi}\cos^{-1}(b) \tag{A.6}
$$

Appendix B

The PDF and CDF of Link Residual Time

B.1 List of the PDF and CDF of Link Residual Time

We derive the PDF and CDF of R_i from (6.3) and (6.4). Recall that R_i is a function of two random variables $\delta_i \sim U(0, \pi)$ and $d_{o,i}$ is linear distributed with the distribution $f_(d_o) = 2d_o/r^2$. Let $s = \sin^{-1}(r/(tv))$, $c = \cos^{-1}(tv/(2r))$, $w(t) = \sqrt{(r^2 - t^2v^2\sin^2(t))}$, $a_1 = 1/\sqrt{(1 - t^2v^2/(4r^2))}$ and $a_2 = 1/\sqrt{(1 - r^2/(tv)^2)}$, then the PDF and CDF of R_i are given by

Case I: $0 \leq t \leq \frac{r}{v}$

$$f_{R_{RN}}(t; v) = \frac{v}{2\pi r a_1} - \frac{v(\frac{(t^2v^2}{2r} + w(c))^2}{2\pi r^3 a_1} + \frac{tv^2}{\pi r^2}\sin(2c) - \frac{2v}{\pi r^2}\int_c^{\pi} \frac{(r^2 - 2t^2v^2\sin^2(t))\cos(t)}{w(t)}dt \tag{B.1}$$

$$F_{R_{RN}}(t; v) = \frac{t^2v^2}{2\pi r^2}\sin(2c) - \frac{2tv}{\pi r^2}\int_c^{\pi} w(t)\cos(t)dt \tag{B.2}$$

Case II: $\frac{r}{v} < t < \frac{2r}{v}\cos\frac{\pi}{4}$

$$f_{R_{RN}}(t; v) = \frac{tv^2}{\pi r^2}\sin(2c) + \frac{4rw(s)\cos(s)}{\pi t r^2 a_2} + \frac{v(1 - r^2(\frac{t^2v^2}{2r} + w(c))^2)}{2\pi r a_1} - \frac{2v}{\pi r^2}\int_0^{s} \frac{(r^2 - 2t^2v^2\sin^2(t))\cos(t)}{w(t)}dt - \frac{2v}{\pi r^2}\int_c^{s} \frac{(r^2 - 2t^2v^2\sin^2(t))\cos(t)}{w(t)}dt \tag{B.3}$$

$$F_{R_{RN}}(t;v) = 1 + \frac{t^2v^2}{2\pi r^2}\sin(2c) - \frac{2tv}{\pi r^2}\int_0^s w(t)\cos(t)dt - \frac{2tv}{\pi r^2}\int_c^s w(t)\cos(t)dt \tag{B.4}$$

Case III: $\frac{2r}{v}\cos\frac{\pi}{4} \leq t \leq \frac{2r}{v}$

$$f_{R_{RN}}(t;v) = \frac{v}{2\pi r a_1} - \frac{v(\frac{t^2v^2}{2r} - w(c))^2}{2\pi r^3 a_1} + \frac{tv^2}{\pi r^2}\sin(2c) - \frac{2v}{\pi r^2}\int_0^c \frac{(r^2 - 2t^2v^2\sin^2(t))\cos(t)}{w(t)}dt \tag{B.5}$$

$$F_{R_{RN}}(t;v) = 1 + \frac{t^2v^2}{2\pi r^2}\sin(2c) - \frac{2tv}{\pi r^2}\int_0^c w(t)\cos(t)dt \tag{B.6}$$

Bibliography

[1] I.F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.

[2] Ming Ma and Yuanyan yang. Clustering and load balancing in hybrid sensor networks with mobile cluster heads. In *Proc. of International Conference on Quality of Service in Heterogeneous Wired / Wireless Networks, ACM*, 2006.

[3] Shu Zhou, Min-You Wu, and Wei Shu. Finding optimal placements for mobile sensors: wireless sensor network topology adjustment. In *Proc. of Circuits and Systems Symposium on Emerging Technologies: Mobile and Wireless Communication, IEEE*, 2004.

[4] Wei Wang, Vikram Srinivasan, and Kee-Chaing Chua. Using mobile relays to prolong the lifetime of wireless sensor networks. In *Proc. of MobiCom'05*, 2005.

[5] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proc. of Hawaii International Conference on System Sciences, IEEE*, 2000.

[6] Ming Ma and Yuanyan Yang. SenCar: An energy-efficient data gathering mechanism for large-scale multihop sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 18(10):1476–1488, 2007.

[7] Ting-Chao Hou and Victor Li. Transmission range control in multihop packet radio networks. *IEEE Transactions on Communications*, 34(1):38–44, 1986.

[8] Leonard Kleinrock and John Silvester. Optimum transmission radii for packet radio networks or why six is a magic number. In *Proc. of National Telecommunications Conference, IEEE*, 1978.

[9] Jia-Chun Kuo and Wanjiun Liao. Hop count distribution of multihop paths in wireless networks with arbitrary node density: Modeling and its applications. *IEEE Transactions on Vehicular Technology*, 56(4):2321–2331, 2007.

[10] Yun Wang, Xiaodong Wang, Demin Wang, and Dharma P. Agrawal. Range-free localization using expected hop progress in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 20(10):1540–1552, 2009.

[11] Deepak Ganesan, Ramesh Govindan, Scott Shenker, and Deborah Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(4):11–25, 2001.

[12] Yinying Yang, Mirela I. Fonoage, and Mihaela Cardei. Improving network lifetime with mobile wireless sensor networks. *Computer Communications*, 33(4):409–419, 2010.

[13] Ryo Sugihara and Rajesh K. Gupta. Optimal speed control of mobile node for data collection in sensor networks. *IEEE Transactions on Mobile Computing*, 9(1):127–139, 2010.

[14] Theodore S. Rappaport. *Wireless Communications Principles and Practice*. Prentice Hall, 2002.

[15] C. Siva Ram Murthy and B. S. Manoj. *Ad Hoc Wireless Networks: Architectures and Protocols*. Prentice Hall, 2004.

[16] D.D. Perkins, H.D. Hughes, and C.B. Owen. Factors affecting the performance of ad hoc networks. In *Proc. of International Conference on Communications, IEEE*, pages 2048–2052, 2002.

[17] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. of MobiCom'98, ACM*, pages 85–97, 1998.

[18] Samir R. Das, Charles E. Perkins, and Elizabeth M. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proc. of INFOCOM'00, IEEE*, pages 3–12, 2000.

[19] S.R. Das, R. Castaneda, Jiangtao Yan, and R. Sengupta. Comparative performance evaluation of routing protocols for mobile, ad hoc networks. In *Proc. of International Conference on Computer Communications and Networks*, pages 153–161, 1998.

[20] Fan Bai, Narayanan Sadagopan, and Ahmed Helmy. Important: a framework to systematically analyze the impact of mobility on performance of routing protocols for ad hoc networks. In *Proc. of INFOCOM'03, IEEE*, pages 825–835, 2003.

[21] Jun ichi Hakoda, Hideyuki Uehara, and Mitsuo Yokoyama. Performance evaluation of mobile ad hoc routing protocols based on link expiration time and load of node. *Electronics and Communications in Japan*, 87(2):2108–2118, 2004.

[22] Chai-Keong Toh. Associativity-based routing for ad hoc mobile networks. *Wireless Personal Communications*, 4(2):103–139, 1997.

[23] R. Dube, C.D. Rais, Kuang-Yeh Wang, and S.K. Tripathi. Signal stability-based adaptive routing (SSA) for ad hoc mobile networks. *Personal Communications, IEEE*, 4(1):36–45, 1997.

[24] William Su, Sung-Ju Lee, and Mario Gerla. Mobility prediction and routing in ad hoc wireless networks. *International Journal of Network Management*, 11(1):3–30, 2001.

[25] Liang Qin and Thomas Kunz. Increasing packet delivery ratio in DSR by link prediction. In *Proc. of the Hawaii International Conference on System Sciences, IEEE*, pages 300–309, 2003.

[26] Prince Samar and Stephen B. Wicker. On the behavior of communication links of a node in a multi-hop mobile environment. In *Proc. of MobiHoc'04, ACM*, pages 145–156, 2004.

[27] Joo-Han Song, V.W.S. Wong, and V.C.M. Leung. Efficient on-demand routing for mobile ad hoc wireless access networks. *IEEE Journal on Selected Areas in Communications*, 22(7):1374–1383, 2004.

[28] Christian Bettstetter. On the minimum node degree and connectivity of a wireless multihop network. In *Proc. of MobiHoc'02, ACM*, pages 80–91, 2002.

[29] Ozan K. Tonguz and Gianluigi Ferrari. Is the number of neighbors in ad hoc wireless networks a good indicator of connectivity? In *Proc. of International Zurich Seminar on Communications: Access-Transmission-Networking, IEEE*, pages 40–43, 2004.

[30] Gianluigi Ferrari and Ozan K. Tonguz. Minimum number of neighbors for fully connected uniform ad hoc wireless networks. In *IEEE Communication Society*, pages 4331–4335, 2004.

[31] Qing Ling and Zhi Tian. Minimum node degree and k-connectivity of a wireless multihop network in bounded area. In *Proc. of Globecom'07, IEEE*, pages 1296–1301, 2007.

[32] Jiang Wu and Douglas R. Stinson. Minimum node degree and k-connectivity for key predistribution schemes and distributed sensor networks. In *Proc. of International Conference on Wireless Network Security, ACM*, pages 119–124. ACM, 2008.

[33] Hideaki Takagi and Leonard Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246–257, 1984.

[34] Feng Xue and P.R. Kumar. The number of neighbors needed for connectivity of wireless networks. *Wireless Networks*, 10(2):169–181, 2004.

[35] Sanquan Song, Dennis Goeckel, and Donald F. Towsley. An improved lower bound to the number of neighbors required for the asymptotic connectivity of ad hoc networks. *IEEE Transactions on Information Theory*, 2005.

[36] Vivek Mhatre and Catherine Rosenberg. Homogeneous vs. heterogeneous clustered sensor networks: A comparative study. In *Proc. of International Conference on Communications, IEEE*, pages 3646–3651, 2004.

[37] Wendi B. Heinzelman, Anantha P. Chandrakasan, and Hari Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communication*, 1(4):660–670, 2002.

[38] Guoliang Xing, Tian Wang, Weijia Jia, and Minming Li. Rendezvous design algorithms for wireless sensor networks with a mobile base station. In *Proc. of MobiHoc'08*, pages 231–240, 2008.

[39] A. T. Erman, L. V. Hoesel, P. Havinga, and J. Wu. Enabling mobility in heterogeneous wireless sensor networks cooperating with UAVs for mission-critical management. *IEEE Wireless Communication*, 15(6):38–46, 2008.

[40] Rahul C. Shah, Sumit Roy, Sushant Jain, and Waylon Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks. In *Proc. of Sensor Network Protocols and Applications, IEEE*, pages 30–41, 2003.

[41] Arnab Chakrabarti, Ashutosh Sabharwal, and Behnaam Aazhang. Using predictable observer mobility for power efficient design of sensor networks. In *Proc. of International Conference on Information Processing in Sensor Networks*, pages 129–145, 2003.

[42] Pritam Baruah and Rahul Urgaonkar. Learning-enforced time domain routing to mobile sinks in wireless sensor fields. In *Proc. of International Conference on Local Computer Networks*, pages 525–532, 2004.

[43] Jun Luo and Jean-Pierre Hubaux. Joint mobility and routing for lifetime elongation in wireless sensor networks. In *Proc. of INFOCOM'05, IEEE*, 2005.

[44] Shashidhar Rao Gandham, Milind Dawande, Ravi Prakash, and S. Venkatesan. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In *Proc. of Globecom'03, IEEE*, 2003.

[45] Arun A. Somasundara, Aman Kansal, David D. Jea, Deborah Estrin, and Mani B. Srivastava. Controllably mobile infrastructure for low energy embedded networks. *IEEE Transactions on Mobile Computing*, 5(8):958–973, 2006.

[46] Santpal Singh Dhillon and Krishnendu Chakrabarty. Sensor placement for effective coverage and surveillance in distributed sensor networks. In *Proc. of International Wireless Communications and Networking Conference, IEEE*, pages 1609–1614, 2003.

[47] Prithwirh Basu and Jason Redi. Movement control algorithms for realization of fault-tolerant ad hoc robot networks. *IEEE Network*, 18(4):36–44, 2004.

[48] Yi Zou and Krishnendu Chakrabarty. Sensor deployment and target localization based on virtual forces. In *Proc. of INFOCOM'03, IEEE*, pages 1293–1303, 2003.

[49] Nojeong Heo and Pramod K. Varshney. Energy-efficient deployment of intelligent mobile sensor networks. *IEEE Transactions on Systems, Man and Cybernetics-Part A: systems and humans*, 35(1):78–92, 2005.

[50] Guiling Wang, Guohong Cao, Tom La Porta, and Wensheng Zhang. Sensor relocation in mobile sensor networks. In *Proc. of INFOCOM'05, IEEE*, pages 2302–2312, 2005.

[51] Guiling Wang, Guohong Cao, and Thomas F. La Porta. Movement-assisted sensor deployment. *IEEE Transactions on Mobile Computing*, 5(6):640–652, 2006.

[52] You-Chiun Wang, Chun-Chi Hu, , and Yu-Chee Tseng. Efficient placement and dispatch of sensors in a wireless sensor network. *IEEE Transactions on Mobile Computing*, 7(2):262–274, 2008.

[53] Isabel Dietrich and Falko Dressler. On the lifetime of wireless sensor networks. *ACM Transactions on Sensor Networks*, 5(1):1–38, 2009.

[54] Jae-Hwan Chang and Leandros Tassiulas. Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(4):609–619, 2004.

[55] Jack Tsai and Tim Moors. A review of multipath routing protocols: From wireless ad hoc to mesh networks. In *Proc. of ACoRN Early Career Researcher Workshop on Wireless Multihop Networking*, 2006.

[56] Bhaskar Krishnamachari, Deborah Estrin, and Stephen B. Wicker. The impact of data aggregation in wireless sensor networks. In *Proc. of International Conference on Distributed Computing Systems*, pages 575–578, 2002.

[57] Razvan Cristescu, Baltasar Beferull-Lozano, and Martin Vetterli. On network correlated data gathering. In *Proc. of INFOCOM'04, IEEE*, 2004.

[58] Vikas Kawadia and P. R. Kumar. Power control and clustering in ad hoc networks. In *Proc. of INFOCOM'03, IEEE*, pages 459–469, 2003.

[59] Ossama Younis and Sonia Fahmy. Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach. In *Proc. of INFOCOM'04, IEEE*, 2004.

[60] Jie Wu and Fei Dai. Mobility-sensitive topology control in mobile ad hoc networks. *IEEE Transactions on Parallel and Distributed Systems*, 17(6):522–535, 2006.

[61] Xiang-Yang Li, Wen-Zhan Song, and Weizhao Wang. A unified energy-efficient topology for unicast and broadcast. In *Proc. of MobiCom'05, ACM*, pages 1–15, 2005.

[62] Yu Wang and Xiang-Yang Li. Distributed spanner with bounded degree for wireless ad hoc networks. *Foundations of Computer Science*, 14(2):183–200, 2003.

[63] Jilei Liu and Baochun Li. Mobilegrid: capacity-aware topology control in mobile ad hoc networks. In *Proc. of International Conference on Computer Communications and Networks*, pages 570–574, 2002.

[64] Qing Cao, Tarek Abdelzaher, Tian He, and John Stankovic. Towards optimal sleep scheduling in sensor networks for rare-event detection. In *Proc. of International Symposium on Information Processing in Sensor Networks Conference, IEEE*, pages 20–27, 2005.

[65] Gang Lu, Narayanan Sadagopan, Bhaskar Krishnamachari, and Ashish Goel. Delay efficient sleep scheduling in wireless sensor networks. In *Proc. of INFOCOM'05, IEEE*, pages 2470–2481, 2005.

[66] M. Ettus. System capacity, latency, and power consumption in multihop-routed SS-CDMA wireless networks. In *Proc. of Radio and Wireless Conference, IEEE*, pages 55–58, 1998.

[67] T. Meng and R. Volkan. Distributed network protocols for wireless communication. In *Proc. of International Symposium on Circuits and Systems Conference, IEEE*, pages 600–603, 1998.

[68] Timothy J. Shepard. A channel access scheme for large dense packet radio networks. In *Proc. of SIGCOMM'96 , ACM*, pages 219–230, 1996.

[69] Suresh Singh, Mike Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proc. of MobiCom'98, ACM*, pages 181–190, 1998.

[70] Z. Maria Wang, Stefano Basagni, Emanuel Melachrinoudis, and Chiara Petrioli. Exploiting sink mobility for maximizing sensor networks lifetime. In *Proc. of Hawaii International Conference on System Sciences, IEEE*, 2005.

[71] Arun A. Somasundara, Aditya Ramamoorthy, and Mani B. Srivastava. Mobile element scheduling with dynamic deadlines. *IEEE Transactions on Mobile Computing*, 6(4):395–410, 2007.

[72] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *Proc. of MobiHoc'04, ACM*, pages 187–198, 2004.

[73] Fatme EI-Moukaddem, Eric Torng, Guoliang Xing, and Sandeep Kulkarni. Mobile relay configuration in data-intensive wireless sensor networks. In *Proc. of Mobile Adhoc and Sensor Systems, IEEE*, pages 80–89, 2009.

[74] Giuseppe Anastasi, Marco Conti, and Mario Francesco. An analytical study of reliable and energy-efficient data collection in sparse sensor networks with mobile relays. In *Proc. of Wireless Sensor Networks, ACM*, pages 199–215, 2009.

[75] Aman Kansal, Arun A. Somasundara, David D. Jea, Mani B. Srivastava, and Deborah Estrin. Intelligent fluid infrastructure for embedded networks. In *Proc. of MobiSys'04, ACM*, pages 111–124, 2004.

[76] Ryo Sugihara and Rajesh K. Gupta. Improving the data delivery latency in sensor networks with controlled mobility. In *Proc. of International Conference on Distributed Computing in Sensor Systems (DCOSS'08), IEEE*, pages 386–399, 2008.

[77] Ryo Sugihara and Rajesh K. Gupta. Optimizing energy-latency trade-off in sensor networks with controlled mobility. In *Proc. of INFOCOM'09, IEEE*, 2009.

[78] Shuai Gao and Hongke Zhang. Energy efficient path-constrained sink navigation in delay-guaranteed wireless sensor networks. *Journal of Networks*, 5(6):658–665, 2010.

[79] E. M. Saad, M H. Awadalla, M. A. Saleh, H. Keshk, and R. R. Darwish. A data gathering algorithm for a mobile sink in large-scale sensor networks. In *Proc. of the International Conference on Wireless and Mobile Communication, IEEE*, pages 207–213, 2008.

[80] Xiwei Zhang, Lili Zhang, Guihai Chen, and Xia Zou. Probabilistic path selection in wireless sensor networks with controlled mobility. In *Proc. of the International Conference on Wireless Communications Signal Processing, IEEE*, pages 1–5, 2009.

[81] Chen Guo, Tao Peng, Shaoyi Xu, Haiming Wang, and Wenbo Wang. Cooperative spectrum sensing with cluster-based architecture in cognitive radio networks. In *Proc. of Vehicular Technology Conference, IEEE*, pages 1–5, 2009.

[82] Li-Chun Wang, Chung-Wei Wang, and Chuan-Ming Liu. Optimal number of clusters in dense wireless sensor networks: A cross-layer approach. *IEEE Transactions on Vehicular Technology*, 58(2):966–976, 2009.

[83] Ming Ma and Yuanyan Yang. Data gathering in wireless sensor networks with mobile collectors. In *Proc. of International Parallel and Distributed Processing Symposium, IEEE*, pages 1–9, 2008.

[84] Arnab Chakrabarti, Ashutosh Sabharwal, and Behnaam Aazhang. Communication power optimization in a sensor network with a path-constrained mobile observer. *ACM Transactions on Sensor Networks*, 2(3):297–324, 2006.

[85] David Jea, Arun Somasundara, and Mani Srivastava. Multiple controlled mobile elements (data mules) for data collection in sensor networks. In *Proc. of International Conference on Distributed Computing in Sensor Systems (DCOSS'05), IEEE*, pages 244–257, 2005.

[86] Jian Ma, Canfeng Chen, and Jyri P. Salomaa. mWSN for large scale mobile sensing. *Signal Processing Systems*, 51(2):195–206, 2008.

[87] Torsha Banerjee, Bin Xie, Jung Hyun Jun, and Dharma P. Agrawal. Increasing lifetime of wireless sensor networks using controllable mobile cluster heads. *Wireless Communications and Mobile Computing*, 10(3):313–336, 2010.

[88] Mirela Marta and Mihaela Cardei. Improved sensor network lifetime with multiple mobile sinks. *Pervasive and Mobile Computing*, 5(5):542–555, 2009.

[89] Ivan Stojmenovic and Xu Lin. Power-aware localized routing in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(11):1122–1133, 2001.

[90] David K. Goldenberg, Jie Lin, A. Stephen Morse, Brad E. Rosen, and Y. Richard Yang. Towards mobility as a network control primitive. In *Proc. of MobiHoc'04, ACM*, 2004.

[91] Chiping Tang and Philip K. McKinley. Energy optimization under informed mobility. *IEEE Transactions on Parallel and Distributed Systems*, 17(9):947–962, 2006.

[92] W. Zhao, M. Ammar, and E. Zegura. Controlling the mobility of multiple data transport ferries in a delay-tolerant network. In *Proc. of INFOCOM'05, IEEE*, pages 1407–1418, 2005.

[93] Paul S. Heckbert. *Graphics Gems IV*. Academic Press, 1994.

[94] Di Ma, Meng Joo Er, and Bang Wang. Analysis of hop-count-based source-to-destination distance estimation in wireless sensor networks with applications in localization. *IEEE Transactions on Vehicular Technology*, 59(6):2998–3011, 2010.

[95] Michele Zorzi and Ramesh R. Rao. Geographic random forwarding (GeRaF) for ad hoc and sensor networks: multihop performance. *IEEE Transactions on Mobile Computing*, 2(4):337–348, 2003.

[96] Leonard E. Miller. Probability of a two-hop connection in a random mobile network. In *Proc. of International Conference on Information Sciences and Systems*, 2001.

[97] C. Bettstetter and J. Eberspacher. Hop distances in homogeneous ad hoc networks. In *Proc. of International Conference on Vehicular Technology, IEEE*, pages 2286–2290, 2003.

[98] Stefan Dulman, Michele Rossi, Paul Havinga, and Michele Zorzi. On the hop count statistics for randomly deployed wireless sensor networks. *International Journal of Sensor Networks*, 1(1/2):89–102, 2006.

[99] Guoqiang Mao, Zijie Zhang, and Brian D.O. Anderson. Probability of k-hop connection under random connection model. *IEEE Communications Letters*, 14(11):1023–1025, 2010.

[100] Ioannis Chatzigiannakis, Athanasios Kinalis, and Sotiris Nikoletseas. Sink mobility protocols for data collection in wireless sensor networks. In *Proc. of*

International Workshop on Mobility Management and Wireless Access, ACM, 2006.

[101] Stefano Basagni, Michele Nati, Chiara Petrioli, and Roberto Petroccia. Rome: Routing over mobile elements in WSNs. In *Proc. of Globecom'09, IEEE*, 2009.

[102] Guojun Wang, Tian Wang, Weijia Jia, Minyi Guo, Hsiao-Hwa Chen, and Mohsen Guizani. Local update-based routing protocol in wireless sensor networks with mobile sinks. In *Proc. of International Conference on Communications, IEEE*, 2007.

[103] Kristóf Fodor and Attila Vidács. Efficient routing to mobile sinks in wireless sensor networks. In *Proc. of International Conference on Wireless Internet, ACM*, pages 1–7, 2007.

[104] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*. Edited by Torrrasz tiehki and Hank Koti, Chapter 5, pages 153-181, Kluwer Academic Publishers, 1996.

[105] M. Naserian, K. E. Tepe, and M. Tarique. Routing overhead analysis for reactive routing protocols in wireless ad hoc networks. In *Proc. of International Conference on Wireless and Mobile Computing, Networking and Communications, IEEE*, pages 22–24, 2005.

[106] Fahimeh Rookhosh, Abolfazl Toroghi Haghighat, and Saeed Nickmanesh. Disjoint categories in low delay and on-demand multipath dynamic source routing ad hoc networks. In *Proc. of International Conference on Distributed Framework and Applications, IEEE*, pages 207–213, 2008.

[107] Asis Nasipuri, Robert Castaeda, and Samir R. Das. Performance of multipath routing for on-demand protocols in mobile ad hoc networks. *Mobile Networks and Applications*, 6(4):339–349, 2001.

[108] Tsung-Chuan Huang and Chi-Chen Chan. Caching strategies for dynamic source routing in mobile ad hoc networks. In *Proc. of International Conference on Wireless Communications and Networking, IEEE*, pages 4239–4243, 2007.

[109] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, 2001.

[110] Hui Li and Dan Yu. A statistical study of neighbor node properties in ad hoc network. In *Proc. of International Conference on Parallel Processing*, pages 103–108, 2002.